

A STATISTICAL MODEL FOR MEASURING ACCURACY AND PRECISION
IN LEO-BASED STAR TRACKERS: AN ANALYSIS OF DISTURBANCE
PROPAGATION

A Thesis presented to
the Faculty of California Polytechnic State University,
San Luis Obispo

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Aerospace Engineering

by
Gagandeep Thapar

June 2023

© 2023

Gagandeep Thapar

ALL RIGHTS RESERVED

COMMITTEE MEMBERSHIP

TITLE: A Statistical Model for Measuring Accuracy and Precision in LEO-based Star Trackers: An Analysis of Disturbance Propagation

AUTHOR: Gagandeep Thapar

DATE SUBMITTED: June 2023

COMMITTEE CHAIR: Kira Abercromby, Ph.D.
Professor of Aerospace Engineering

COMMITTEE MEMBER: Leo Torres, Ph.D.
Professor of Aerospace Engineering

COMMITTEE MEMBER: John Bellardo, Ph.D.
Professor of Computer Science

COMMITTEE MEMBER: Eric Mehiel, Ph.D.
Professor of Aerospace Engineering

ABSTRACT

A Statistical Model for Measuring Accuracy and Precision in LEO-based Star Trackers: An Analysis of Disturbance Propagation

Gagandeep Thapar

As the complexity of CubeSat missions continues to escalate, the usage of star trackers as reliable guidance systems has garnered widespread interest. This increased demand, coupled with prohibitive costs from commercial vendors, has made the development of in-house star trackers a compelling alternative. However, the high sensitivity of star trackers to various sources of error poses a significant challenge, especially given the lack of existing tools to estimate the accuracy of a given configuration. This thesis presents a novel approach to address this gap by introducing a statistical model capable of predicting the accuracy of star trackers via replication of the underlying measurement process and error models derived from first principles. The study examines the associations between star tracker accuracy and sensor-specific factors such as image plane translation, rotation, and centroiding precision and extends to assessing the impact of the thermal and radiation components of the LEO environment on star tracker accuracy throughout its operational lifespan with respect to their effects on additional distortion and image noise, respectively. It is determined that deviations in image plane translation result in a consistent bias proportional to the magnitude of lateral aberration on accuracy while rotations in the image plane introduce considerable error in attitude estimates. The influence of centroiding precision and image noise on star vector determination and star detection, respectively, and their relationship with overall accuracy is also examined based on experimental and simulation data.

ACKNOWLEDGMENTS

They say it takes a village to raise a child, and I've found that adage to be completely true. First and foremost, I want to thank my family for their unconditional love and support. Your belief in me has been the driving force behind my career and personal growth. Without your encouragement, I would not have had the opportunity to pursue my passions and become the person I am today. To Dr. Torres, whose relentless optimism and invaluable guidance have made a significant difference. You've been an exceptional advisor and I appreciate just being able to bounce ideas off you and explore various fields and concepts. This thesis would not be what it is without your support, and I am truly grateful for the time and effort you have invested in me. To Dr. Abercromby, for your outstanding enthusiasm. Your passion for guidance and navigation has been truly inspiring and it played a crucial role in shaping my passion and decision to pursue a career in GNC. Your contagious excitement has motivated me to push the boundaries and strive for excellence and I hope you continue to instill that for your future students. To Dr. Mehiel and Dr. Bellardo, I want to extend a thank you for your support throughout my time at Cal Poly and during the completion of this thesis. Your guidance and encouragement have instilled in me a sense of going above and beyond. I am grateful for the opportunity to step out of my comfort zone and learn new things under your mentorship. Lastly, I would like to express my thanks to the entire Cal Poly Aerospace Department and the Cal Poly CubeSat Laboratory. The learning environment provided by the department and the opportunities offered by the laboratory have been invaluable. Cal Poly has not only taught me the importance of "Learn by Doing" but has also encouraged me to be interested, ask questions, and seek answers. I'm grateful for the chance to learn from my peers and broaden my horizons.

TABLE OF CONTENTS

	Page
LIST OF TABLES	viii
LIST OF FIGURES	ix
CHAPTER	
1 Introduction	1
1.1 Statement of Problem	1
1.2 Objective	3
2 Background	4
2.1 Evolution of CubeSat Pointing Requirements	4
2.2 Importance of Work	6
2.3 Attitude Determination	8
2.3.1 Reference Frames	9
2.3.2 Rotation Matrices	11
2.3.3 Quaternions	13
2.3.4 Wahba's Problem	15
3 Star Tracker Operation	18
3.1 Star Catalog	19
3.2 Hardware Interfaces	23
3.3 Centroiding	24
3.4 Star Identification	28
3.5 Operating Modes	29
3.6 Space Environment	29
3.6.1 Thermal Environment	30
3.6.2 Radiation Environment	31
4 Measurement Process Model	33
4.1 Model Overview	33
4.2 Error Modeling	41
4.2.1 Hardware Analysis	42
4.2.2 Noise Simulation	46

4.2.3	Centroiding Accuracy	61
4.3	Effects of the Space Environment	65
4.3.1	Thermal Environment	65
4.3.2	Radiation Environment	67
4.4	QUEST	68
4.5	Gamma Function	70
5	Sensitivity and Multivariate Analysis	76
5.1	Analysis Approach	76
5.2	Multivariate Analysis	77
5.3	Special Case Analysis	98
6	Results and Conclusion	103
6.1	Conclusion	103
6.2	Future Work	105
	BIBLIOGRAPHY	108
	APPENDICES	
A	Hardware Aberration Model	113
B	Connected Component Algorithm	115

LIST OF TABLES

Table	Page
3.1 Sample star catalog entry from Yale Bright Star Catalog	20
4.1 Model star tracker properties used for analysis	73

LIST OF FIGURES

Figure	Page
3.1 High-level measurement process of a star tracker	18
3.2 Celestial sphere (above) and Plane (below) filtered from the Yale Bright Star Catalog (BSC5) on stars with apparent magnitude 5.2 and less; $(\alpha, \delta) \equiv (46.25, 33.54)^\circ$	21
3.3 Pinhole projection model	26
4.1 Error sources and propagation within the star tracker measurement process	42
4.2 Unblurred star (above) and blurred star (below) by means of a Gaussian Filter with $\sigma = 2$, $\text{ROI} \in \mathbf{R}^{7 \times 7}$	50
4.3 Sample simulated image with no noise; region of interest $\in \mathbf{R}^{21 \times 21}$, $\sigma = 5$ for visibility	51
4.4 Sample simulated image with light and dark noise superimposed; region of interest $\in \mathbf{R}^{21 \times 21}$, $\sigma = 5$ for visibility	55
4.5 Sample simulated image with conversion noise superimposed; region of interest $\in \mathbf{R}^{21 \times 21}$, $\sigma = 5$ for visibility	58
4.6 Sample simulated image with threshold applied; region of interest $\in \mathbf{R}^{21 \times 21}$, $\sigma = 5$ for visibility	59
4.7 Before (above) and after (below) applying the brightness threshold; $\text{ROI} \in \mathbf{R}^{21 \times 21}$, $\sigma = 5$ and signal boosted by factor of 5 for visibility	60
4.8 Centroid accuracy versus number of electrons with Gaussian filtering $\sigma=2$	62
4.9 Centroid accuracy versus apparent magnitude, $R^2 = 0.9989$	64
5.1 Attitude accuracy for a star tracker with ideal parameters	77
5.2 Attitude accuracy (11.601 ± 28.376 arcseconds, 1σ) with all inputs varied	78
5.3 Attitude accuracy (6.053 ± 4.086 arcseconds, 1σ) versus 3 degrees of image plane translation	79
5.4 Attitude accuracy (2.804 ± 33.301 arcseconds, 1σ) versus 3 degree image plane rotation	80
5.5 Attitude accuracy (6.266 ± 3.999 arcseconds, 1σ) versus lateral image plane translation	81

5.6	Attitude accuracy (0.000 ± 0.630 arcseconds, 1σ) versus vertical image plane translation	82
5.7	Sensitivity of accuracy with respect to lateral image plane translation	83
5.8	Sensitivity of accuracy with respect to vertical image plane translation	84
5.9	Attitude accuracy (0.097 ± 0.329 arcseconds, 1σ) versus image plane rotation about lateral axes	85
5.10	Attitude accuracy (0.002 ± 35.877 arcseconds, 1σ) versus image plane rotation about vertical axis	86
5.11	Sensitivity of accuracy with respect to image plane rotation about lateral axes	87
5.12	Sensitivity of accuracy with respect to image plane rotation about vertical axis	88
5.13	Attitude accuracy (0.401 ± 2.814 arcseconds, 1σ) versus centroid accuracy	90
5.14	Sensitivity of accuracy with respect to centroid accuracy	91
5.15	Attitude accuracy (0.001 ± 0.005 arcseconds, 1σ) versus least-squares vector regression centroid accuracy	92
5.16	Star quantity-based failure rate with respect to maximum magnitude star discernable	94
5.17	Sensitivity of accuracy with respect to maximum magnitude star discernable	95
5.18	Attitude accuracy (12.515 ± 28.123 arcseconds, 1σ) with effects from the thermal environment	96
5.19	Attitude accuracy (11.206 ± 28.856 arcseconds, 1σ) with effects from the radiation environment	97
5.20	Attitude accuracy (0.823 ± 3.329 arcseconds, 1σ) of fully-calibrated star tracker model	99
5.21	Attitude accuracy (0.090 ± 36.280 arcseconds, 1σ) of translation-calibrated star tracker model	100
5.22	Attitude accuracy (6.735 ± 5.089 arcseconds, 1σ) of rotation-calibrated star tracker model	101
5.23	Attitude accuracy (9.424 ± 29.725 arcseconds, 1σ) of improved-centroiding star tracker model	102

List of Algorithms

1	Process for Star Projection Determination	22
2	Process of a Simulation Run	35
3	Simulation Architecture	36
4	QUaternion ESTimator Algorithm	69
5	Measurement Process Model	75
6	Connected Component Algorithm	115

Chapter 1

INTRODUCTION

1.1 Statement of Problem

CubeSats, developed in 1999 by Dr. Jordi Puig-Suari and Dr. Bob Twiggs, were initially designed to reduce costs and development time while increasing the accessibility to space and sustain frequent launches [1]. They typically use commercial-off-the-shelf, or, COTS, components to remain affordable [1] while still meeting mission requirements and have become the standard for several university programs and commercial businesses [1][2][3] for a wide range of space applications, including Earth observation, technology demonstration, and scientific research.

Since its inception in 1999, CubeSat technology has greatly matured, enabling them for more sophisticated operations. The growing complexity of CubeSat missions has increased the demand for more accurate attitude determination solutions. CubeSats are now being used for a wide range of applications, from monitoring environmental conditions to conducting scientific experiments in space. To perform these tasks, CubeSats need to be precisely oriented in space to point their payloads in the correct direction. Attitude determination is the process of determining the orientation of a spacecraft in three-dimensional space. Accurate attitude determination is essential for the proper functioning of many CubeSat payloads, including cameras, telescopes, spectrometers, and other scientific instruments.

NASA's CubeSat Infrared Laser CrosslinK (CLICK) mission is a prime example of the need for accurate attitude determination in CubeSats. As a technology demonstration of optical communications, the mission requires precise pointing of the CubeSats' communication antennas to enable successful laser-based transmissions be-

tween spacecraft. Conservative estimates suggest that the system requires within 5 arcseconds of control accuracy [4] to achieve its goals, emphasizing the critical role of attitude determination. Given the limitations of traditional attitude determination techniques for CubeSats, the development of innovative and cost-effective solutions is crucial to enable future missions such as CLICK and expand the potential of CubeSats for scientific research and space exploration.

The star tracker is an attitude determination sensor that CubeSats employ for their high accuracy and compliance to the CubeSat standard. This sensor uses a camera to capture images of star fields and compares them with a star catalog to determine the spacecraft's orientation in space. The star tracker's ability to provide accurate and reliable attitude estimations has made it a popular choice for CubeSats, especially those requiring precise pointing for scientific and technological applications. Star trackers, however, can be cost-prohibitive for some CubeSat developers, particularly those at the university level with limited budgets. The high cost of the hardware and software required for a star tracker can make it challenging for these developers to implement the technology in their CubeSat missions. Additionally, star trackers are highly sensitive to errors in both hardware and software, which can negatively affect their performance. The sensitivity of star trackers to the space environment, such as the thermal environment and radiation, can also lead to errors in the attitude estimation, further adding to the complexity of their operation. Therefore, while star trackers remain an attractive attitude determination solution for CubeSats, their cost and sensitivity to errors and space environment require careful consideration by CubeSat developers.

1.2 Objective

The lack of a reliable and cost-effective attitude determination system has been a significant limitation on the full potential of CubeSats. The goal of this thesis is to analyze the error sources in the star tracker measurement model and their impact on the final attitude estimation, enabling developers, especially at the university level, to further understand and characterize their own star tracker. The effects of hardware deviations, sensor noise, software performance, and the space environment, specifically the thermal and radiation environments, will be examined. This thesis will delve into the operation of a star tracker and form a framework based on the measurement process to provide attitude estimations and will focus on how errors can occur in the star tracker system, why these errors can impact the measurement process, and how they can propagate to distort the final attitude estimate. By understanding the sources of error in the star tracker system, this thesis aims to provide developers with insights into how different factors can affect the accuracy of the star tracker to better prioritize their efforts and ultimately enhance the performance and reliability of the star tracker the CubeSat missions rely on for attitude determination. With this research, this thesis hopes to contribute to the development of innovative and cost-effective attitude determination solutions for CubeSats, advancing the potential of these small spacecraft for scientific research and space exploration.

Chapter 2

BACKGROUND

2.1 Evolution of CubeSat Pointing Requirements

Small satellites, especially CubeSats, have become increasingly popular in recent years due to their low cost, fast development times, and versatility. It can be seen, however, that the pointing requirements, specifically requirements for attitude determination, have evolved over time for CubeSats and have become increasingly stricter. Developing low-cost star trackers are important for two main reasons: they enable CubeSats to be capable for more complex missions, and they lower the barrier of entry to space for a large proportion of the CubeSat community - especially at the university level. Star trackers enable CubeSats to close the gap between more complex spacecraft at a fraction of the development time and cost.

The first CubeSats were launched in the early 2000s and were primarily used for technology demonstration and educational purposes. At this time, the pointing requirements for CubeSats were not particularly strict, and the satellites were often allowed to drift in their orbits without much concern for their attitude [5]. However, as CubeSats began to be used for scientific and commercial applications, the requirements for their pointing accuracy began to increase. Attitude determination was typically achieved using low-cost magnetometers [2], horizon sensors (0.02° of accuracy [6]) and sun sensors (0.5° of accuracy [7]). These sensors were relatively inaccurate and provided only a rough estimate of the satellite's orientation and were sufficient for the time as most CubeSat-based missions did not need specific attitude information to achieve their objective other than to communicate with their ground-station such as the Dampening and Vibrations Experiment, or DAVE, developed by

the Cal Poly CubeSat Lab [2]. However, as CubeSats began to be used for more demanding applications, such as Earth observation and remote sensing, the pointing requirements became much stricter. In order to achieve the required pointing accuracy, CubeSats began to incorporate more advanced attitude determination sensors, namely star trackers.

Laser communication, also known as optical communication, in CubeSats represents a significant advancement in space communication technologies. The commitment of the industry, particularly NASA's Jet Propulsion Lab (JPL), to developing laser communication systems for CubeSats is clear. A JPL study sought to identify the most promising development paths for optical communications in CubeSats with the intent of enabling missions that span from near-Earth to deep-space, working to mitigate key risks associated with this technology. Their ambition is to ensure the viability and competitiveness of optical communications as an alternative to traditional RF communications for CubeSats [5]. Traditional CubeSats have relied on radio frequency communications, which are limited in terms of bandwidth. In contrast, laser communication can achieve much higher data rates, making it ideal for tasks that involve transmitting large volumes of data, such as high-resolution imaging and scientific data transmission. However, the implementation of laser communication in CubeSats introduces unique challenges. One of the primary requirements is the need for very precise pointing control. To maintain line-of-sight for successful optical communication, the system must be able to point with extreme accuracy, often on the order of 4 arcseconds [4][5]. Such precision necessitates advanced onboard attitude determination systems. The requirement for tight pointing knowledge underscores the complexity of laser communication implementation, as any misalignment can lead to loss of signal, thereby disrupting communication. Therefore, CubeSats employing this technology need to be equipped with sophisticated sensor systems to maintain this stringent level of accuracy.

The pointing requirements for SmallSats/CubeSats have evolved significantly over time, from relatively lax requirements in the early days of CubeSat missions to much stricter requirements for modern scientific and commercial applications. As CubeSats continue to be used for more demanding applications, it is likely that the requirements for their pointing accuracy will continue to increase. However, advances in attitude determination sensors will also make it increasingly possible to achieve these more stringent requirements while simultaneously lowering the barrier to entry for many organizations. Star trackers are extremely cost prohibitive and can start at \$35,000USD [8] and be as expensive as \$120,000USD [9] based on the advertised performance and flight heritage. Because star trackers require relatively minimal amount of hardware, typically just a camera, dedicated processor, and some supporting components, it is often alluring for budget-constrained missions to design and build a custom star tracker [10]. While a valid option for many organizations, it is imperative to understand how and where errors can occur during the measurement process and their effect on the accuracy of the star tracker. By analyzing how each specific parameter affects accuracy, star tracker developers can better prioritize development efforts to maximize improvement with minimum time and cost.

2.2 Importance of Work

By analyzing error propagation, additional insights about the system can be obtained and a tool can be developed for CubeSat developers. A comprehensive mathematical model of the star tracker, encompassing various factors that contribute to errors, can be deduced by examining each physical step of the tracker. Identifying flaws in camera construction, such as the boresight vector normality or focal array tilt, allows for characterization and modeling of immediate errors. Considering different effects, including camera hardware, noise, algorithmic errors, and environmental errors in low

Earth orbit (LEO), a model can be created to estimate the accuracy of a theoretical star tracker based on a given set of inputs. This model can help CubeSat developers assess whether a low-cost star tracker would be sufficient for their mission without the need for experimental testing using mission parameters.

Studying error propagation also helps identify areas where errors are more likely to occur and where it is worthwhile to optimize processes. By fully understanding and parameterizing the star tracker process, it becomes possible to model tracker accuracy in on-orbit conditions and make informed decisions regarding attitude determination sensor selection, even without prior experimental information. Alternative methods for cost reduction in star trackers include hardware optimization and custom firmware development for the optical module. However, these approaches still entail high development costs. Optimizing the hardware requires knowledge of how the optical system functions and how different hardware choices affect accuracy estimates. The star tracker model developed through error propagation analysis eliminates the guesswork in hardware selection and reduces development costs and time. Software optimization is another option, but it can be challenging and expensive without a deep understanding of the physics, image analysis process, and flight software. Analyzing error propagation allows us to identify error sources and determine how hardware selection can mitigate them. This analysis also provides insights into selectively accepting errors by choosing affordable hardware, thereby reducing overall star tracker costs.

Furthermore, the tool developed through the analysis of error propagation in star trackers can also find valuable application within larger spacecraft or CubeSat simulations, enhancing the fidelity of the Attitude Determination and Control System (ADCS). Simulations play a crucial role in spacecraft design, mission planning, and operational testing. They allow for the evaluation of different configurations, algorithms, and hardware choices without the need for expensive and time-consuming

physical prototypes. Accurately modeling the ADCS system is particularly important as it directly influences the spacecraft's attitude determination, control, and stability. By incorporating this tool, engineers can create more accurate representations of the ADCS system, leading to improved spacecraft design, mission planning, and operational testing. The tool's integration enhances the fidelity of simulations, reduces reliance on physical prototypes, and facilitates a more informed decision-making process for ADCS optimization. They can investigate different hardware configurations, evaluate the performance of different algorithms, and validate the suitability of various attitude determination sensors within the simulation environment. This holistic approach ensures that the ADCS system is optimized to meet the mission requirements and enhances the overall reliability and mission success. By incorporating this tool into simulations, engineers and mission planners can achieve a more accurate representation of the spacecraft's behavior and performance in various scenarios. Given the rapid progress of CubeSat technology and the need for equally capable attitude determination sensors within limited budgets, it is evident that a demand exists for such sensors to meet the evolving capabilities of CubeSats.

2.3 Attitude Determination

Attitude Determination is the ability to determine the orientation of one reference frame, e.g., the body-fixed reference frame of a spacecraft, to an inertial reference frame, e.g., the Earth-Centered Inertial (ECI) Reference Frame. While this transformation can be represented in several methods such as via Euler Angles, i.e., three angles defined off a coordinate frame to specify a direction, the most common forms are quaternions and rotation matrices [11].

2.3.1 Reference Frames

A 3-Dimensional reference frame, $\vec{\mathcal{F}}$, is a triad of principal axes (e.g., as presented in Equation 2.1) which form a dextral-orthogonal basis, by convention, such that any physical vector, $\vec{\mathbf{r}}$, can be described as a linear combination of the unit vectors in $\vec{\mathcal{F}}$ as in 2.3 [12].

$$\vec{\mathcal{F}} = \begin{bmatrix} \vec{\mathbf{x}} \\ \vec{\mathbf{y}} \\ \vec{\mathbf{z}} \end{bmatrix} \quad (2.1)$$

$$\vec{\mathbf{r}} = r_{x,1} \vec{\mathbf{x}} + r_{y,1} \vec{\mathbf{y}} + r_{z,1} \vec{\mathbf{z}} \quad (2.2)$$

$$\vec{\mathbf{r}} = \vec{\mathcal{F}}_1^T \mathbf{r}_1 \quad (2.3)$$

It should be noted that the same physical vector, $\vec{\mathbf{r}}$, will have different representations in different reference frames, $\vec{\mathcal{F}}_1$, and $\vec{\mathcal{F}}_2$, e.g.,

$$\vec{\mathbf{r}} = \vec{\mathcal{F}}_1^T \mathbf{r}_1 = \vec{\mathcal{F}}_2^T \mathbf{r}_2$$

but still describe the same information, $\vec{\mathbf{r}}$.

Transformations between $\vec{\mathcal{F}}_1$ and $\vec{\mathcal{F}}_2$ can be represented by the dot product of their respective axes and, in practice, represent how a vector in one reference frame can be represented in a different reference frame [12]. This transformation is outlined below where the representation \mathbf{r}_1 is being transformed to match the representation of \mathbf{r}_2 .

$$\vec{\mathcal{F}}_2^T \mathbf{r}_2 = \vec{\mathcal{F}}_1^T \mathbf{r}_1 \quad (2.4)$$

$$\vec{\mathcal{F}}_2 \cdot \vec{\mathcal{F}}_2^T \mathbf{r}_2 = \vec{\mathcal{F}}_2 \cdot \vec{\mathcal{F}}_1^T \mathbf{r}_1 \quad (2.5)$$

$$\mathbf{r}_2 = \vec{\mathcal{F}}_2 \cdot \vec{\mathcal{F}}_1^T \mathbf{r}_1 \quad (2.6)$$

$$\mathbf{r}_2 = \mathbf{C}_{21} \mathbf{r}_1 \quad (2.7)$$

$$\mathbf{C}_{21} = \vec{\mathcal{F}}_2 \cdot \vec{\mathcal{F}}_1^T \quad (2.8)$$

In the context of attitude determination, the physical vector represents the attitude, or pointing direction, of the spacecraft, the reference frames, $\vec{\mathcal{F}}_1$ and $\vec{\mathcal{F}}_2$, represent the attitude determining sensor reference frame (e.g., the reference frame of the star tracker) and the inertial reference frame, respectively, and the rotation matrix, \mathbf{C}_{21} , represents the attitude determined from the sensor. The aim of an attitude determination sensor is to compute the rotation, \mathbf{C}_{21} , as accurately as possible with limited knowledge.

While the attitude determining sensor provides the rotation from the inertial frame to itself, in practice, a third reference frame can be utilized, e.g., the spacecraft body-fixed reference frame, to provide more practical information. This reference frame describes the spacecraft bus and is useful to use during the design of the spacecraft. For the purposes of this work, the attitude determining reference frame and the spacecraft body-fixed reference frame are parallel and therefore interchangeable. It will be referred to as the body-fixed reference frame or the camera reference frame.

The Earth-Centered Inertial Reference Frame (ECI frame) is typically used to represent the inertial frame for attitude determination problems. The ECI frame forms its basis with its \vec{x} axis pointing towards the First Point of Aries, its \vec{z} axis pointing through the North Pole, and its \vec{y} axis completing the triad to form the dextral orthogonal basis [12]. This formulation of the body-fixed reference frame and the ECI frame will be frequently used in understanding how the star tracker operates.

2.3.2 Rotation Matrices

A rotation matrix, as described in 2.8, is a square matrix whose size is equivalent to the number of rotating axes in the system, typically 3. Rotation matrices have a few special properties which are integral to its implementation, the first being that the transpose of a rotation matrix \mathbf{C}_{ab} , \mathbf{C}_{ab}^T , represents the opposing rotation described by \mathbf{C}_{ab} ; that is,

$$\begin{aligned}\mathbf{r}_a &= \mathbf{C}_{ab}\mathbf{r}_b \\ \mathbf{C}_{ab}^T\mathbf{r}_a &= \mathbf{C}_{ab}^T\mathbf{C}_{ab}\mathbf{r}_b \\ \mathbf{C}_{ab}^T\mathbf{r}_a &= \mathbf{r}_b\end{aligned}$$

and can be used to identify the rotation from $\vec{\mathcal{F}}_a$ to $\vec{\mathcal{F}}_b$. In conclusion, \mathbf{C}_{ab}^T is equivalent to \mathbf{C}_{ba} . It can also be seen that the product of \mathbf{C}_{ab} and \mathbf{C}_{ba} is equivalent to the identity matrix, $\mathbf{1}^{n \times n}$. Notice, if a rotation matrix is said to be the identity matrix, $\mathbf{1}$, this implies there is no rotation, or that the rotation angles are 0, as the resulting vector when the rotation matrix is applied will be itself. This will be important to validate composed rotation matrices.

Another key concept is that rotation matrices can also be chained together to represent rotations in multiple axes successively. Suppose there exists three reference frames, $\vec{\mathcal{F}}_1$, $\vec{\mathcal{F}}_2$, and $\vec{\mathcal{F}}_3$ such that

$$\vec{\mathbf{r}} = \vec{\mathcal{F}}_1\mathbf{r}_1 = \vec{\mathcal{F}}_2\mathbf{r}_2 = \vec{\mathcal{F}}_3\mathbf{r}_3$$

Suppose the rotation from $\vec{\mathcal{F}}_1$ to $\vec{\mathcal{F}}_2$ is known to be \mathbf{C}_{21} and the rotation from $\vec{\mathcal{F}}_2$ to $\vec{\mathcal{F}}_3$ is known to be \mathbf{C}_{32} . It can be shown that the rotation from $\vec{\mathcal{F}}_1$ to $\vec{\mathcal{F}}_3$, \mathbf{C}_{31} ,

can be computed directly as shown in 2.14.

$$\mathbf{r}_2 = \mathbf{C}_{21}\mathbf{r}_1 \quad (2.9)$$

$$\mathbf{r}_3 = \mathbf{C}_{32}\mathbf{r}_2 \quad (2.10)$$

$$\mathbf{r}_3 = \mathbf{C}_{32}(\mathbf{C}_{21}\mathbf{r}_1) \quad (2.11)$$

$$\mathbf{r}_3 = (\mathbf{C}_{32}\mathbf{C}_{21})\mathbf{r}_1 \quad (2.12)$$

$$\mathbf{r}_3 = \mathbf{C}_{31}\mathbf{r}_1 \quad (2.13)$$

$$\mathbf{C}_{31} = \mathbf{C}_{32}\mathbf{C}_{21} \quad (2.14)$$

Notice, because matrix products are not commutative, i.e., $\mathbf{AB} \neq \mathbf{BA}$, it is imperative the rotations are applied in the correct order.

$$\mathbf{C}_{32}\mathbf{C}_{21} \neq \mathbf{C}_{21}\mathbf{C}_{32}$$

While the aforementioned properties apply to any rotation matrix, there also exist three special rotation matrices which represent rotations about the principal axes, \vec{x} , \vec{y} , and \vec{z} as presented in 2.15 - 2.17 [12]. The angles, ϕ , θ , and ψ represent the magnitude of the rotation and the \mathbf{R}_{\square} notation represents about which principal axis the rotation matrix revolves around.

$$\mathbf{R}_{\vec{x}}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \quad (2.15)$$

$$\mathbf{R}_{\vec{y}}(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (2.16)$$

$$\mathbf{R}_{\vec{\mathbf{z}}}(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.17)$$

2.3.3 Quaternions

One deficiency of using Euler Angles or rotation matrices is that there exists a singularity, i.e., a specific set of rotations or angles for which the rotation cannot be uniquely determined [12]. This is often called “Gimbal Lock” and can provide computational problems during its determination. To escape this, an additional parameter must be introduced for redundancy. A quaternion, \mathbf{q} , is a 4-dimensional hypervector containing one real part, η , and 3 vector parts, $\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, \epsilon_3]$; i.e.,

$$\mathbf{q} = \begin{bmatrix} \boldsymbol{\epsilon} \\ \eta \end{bmatrix} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \eta \end{bmatrix} \quad (2.18)$$

which can uniquely determine a rotation from a reference frame, much like its rotation matrix counterpart.

Quaternions also embody special properties that must be respected when representing rotations. The first property is that quaternions shall satisfy the unit magnitude constraint; that is, the norm of the quaternion shall be 1.

$$\begin{aligned} \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} + \eta^2 &= 1 \\ \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 + \eta^2 &= 1 \end{aligned}$$

Successive rotations are also possible with quaternions via quaternion multiplication. Quaternion multiplication is unique due to its composition of a real part and vector part. Similar to how $\mathbf{C}_{31} = \mathbf{C}_{32}\mathbf{C}_{21}$, the quaternion form is represented as $\mathbf{q}_{31} = \mathbf{q}_{32} \otimes \mathbf{q}_{21}$ where \otimes denotes quaternion multiplication. Quaternion multiplication is defined in Equation 2.19

$$\mathbf{q}_3 = \mathbf{q}_1 \otimes \mathbf{q}_2 \quad (2.19)$$

$$\boldsymbol{\epsilon}_3 = \eta_2 \boldsymbol{\epsilon}_1 + \eta_1 \boldsymbol{\epsilon}_2 + \boldsymbol{\epsilon}_1^\times \boldsymbol{\epsilon}_2 \quad (2.20)$$

$$\eta_3 = \eta_1 \eta_2 - \boldsymbol{\epsilon}_2^T \boldsymbol{\epsilon}_1 \quad (2.21)$$

where the \square^\times operator denotes the skew-symmetric operation such that, for a 3-Dimensional vector, $v = [v_x, v_y, v_z]^T$,

$$v^\times = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix} \quad (2.22)$$

Notice, it holds true that for 2 vectors, v and w , the cross product of v and w , $v \times w$, is equivalent to $v^\times w$.

A useful property is that a rotation matrix can be derived from a given quaternion as presented in 2.23 [12]

$$\mathbf{C} = (2\eta^2 - 1) \mathbf{1} + 2\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T - 2\eta\boldsymbol{\epsilon}^\times \quad (2.23)$$

Similar to 2.23, there also exists a method to derive the quaternion from its rotation matrix as presented in 2.24 [12]

$$\eta = \pm \frac{(\text{tr}[\mathbf{C}] + 1)^{\frac{1}{2}}}{2} \quad (2.24)$$

$$\epsilon_1 = \frac{C_{23} - C_{32}}{4\eta} \quad (2.25)$$

$$\epsilon_2 = \frac{C_{31} - C_{13}}{4\eta} \quad (2.26)$$

$$\epsilon_3 = \frac{C_{12} - C_{21}}{4\eta} \quad (2.27)$$

where the $\text{tr}[\square]$ operator symbolizes the trace of the argument and C_{ab} represents the element at row a and column b of matrix \mathbf{C} .

Furthermore, the inverse of a quaternion, \mathbf{q}^* is said to be its conjugate where $\boldsymbol{\epsilon}^* = -\boldsymbol{\epsilon}$. The difference between two quaternions, \mathbf{q}_Δ , is said to be the product of one quaternion and the conjugate of the other as presented in 2.28 and the angular separation is defined by 2.29 [13] where **arctan2** is a modified **arctan** function to remove quadrant ambiguity.

$$\mathbf{q}_\Delta = \mathbf{q}_2 \otimes \mathbf{q}_1^* \quad (2.28)$$

$$\theta_\Delta = 2 \cdot \text{arctan} 2 \left(\sqrt{\boldsymbol{\epsilon}_\Delta^T \boldsymbol{\epsilon}_\Delta}, \eta_\Delta \right) \quad (2.29)$$

The angular difference, θ_Δ , represents the error between \mathbf{q}_1 and \mathbf{q}_2 in angular units.

It is important to notice that the angular separation is not equivalent to the norm of the set of rotation angles; i.e., for $\mathbf{C} = \mathbf{R}_x(\phi)\mathbf{R}_y(\theta)\mathbf{R}_z(\psi)$, $\theta_\Delta \neq \sqrt{\phi^2 + \theta^2 + \psi^2}$. The expressions described in 2.29, 2.23, and 2.24 will be very useful in determining rotations and measuring accuracy when discussing the effects of errors on attitude estimation.

2.3.4 Wahba's Problem

The goal of the star tracker is to estimate the rotation from ECI to the body-fixed frame which can be computed by solving Wahba's Problem. Wahba's Problem, presented in Equation 2.30, is an optimization problem designed to find a rotation, \mathbf{C} ,

between a set of vectors in two reference frames by minimizing the residual error defined as the norm of the difference between the vector representations in one reference frame and the vector representations in the other rotated by \mathbf{C} [14].

$$J(\mathbf{C}) = \frac{1}{2} \sum_{k=1}^N a_k \|\mathbf{w}_k - \mathbf{C}\mathbf{v}_k\|^2 \quad (2.30)$$

where $\mathbf{C} \in \mathbf{R}^{3 \times 3}$ represents the rotation, N is the number of vectors in each frame, a_k represents the weight or uncertainty of its corresponding vector, \mathbf{w}_k represents the physical vector's representation in one frame (i.e., the body-fixed frame), and \mathbf{v}_k represents the physical vector's representation in the other frame (i.e., the ECI frame). When the rotation matrix, \mathbf{C} , is chosen such that the cost function is minimized, i.e., the residual error between the vectors is minimized, it can be said that \mathbf{C} is the “optimal” rotation of the system with respect to the residual of the error. It should be noted that Wahba’s Problem requires there be at least 2 physical vectors in the system as there exist an infinite number of rotations for a single vector. Several solutions to Wahba’s Problem exist such as Davenport’s q-method, TRIAD, and singular value decomposition presented in Equation 2.31 [15]. One particular algorithm, however, has been increasingly popular for its minimal computation requirements while maximizing accuracy [11]. The QUaternion ESTimator Algorithm, or QUEST Algorithm, is popular for resource-constrained spacecraft and allows for an optional set of weights to assign uncertainty to each pair of vectors; a useful tool for sensor fusion applications. In the case of the star tracker, the weights can be assigned according to the apparent magnitude of the star, and therefore the accuracy of its centroid, or can be kept equal. Algorithm 4 presents the algorithm for computing the optimal quaternion, \mathbf{q}_{est} by means of the QUEST Algorithm.

$$\mathbf{B} = \sum_{k=1}^n a_k \mathbf{w}_k \mathbf{v}_k^T \quad (2.31)$$

$$\mathbf{B} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (\text{Singular Value Decomposition}) \quad (2.32)$$

$$\mathbf{M} = \text{diag}([1 \ 1 \ \det(\mathbf{U})\det(\mathbf{V})]) \quad (2.33)$$

$$\mathbf{C} = \mathbf{U}\mathbf{M}\mathbf{V}^T \quad (2.34)$$

The solution to Wahba's Problem presented in Equation 2.31 is typically not used as singular value decomposition requires high computational resources.

Chapter 3

STAR TRACKER OPERATION

A star tracker, at its essence, functions as an optical sensor, often employing a monochromatic camera, that captures images of the celestial sphere with the purpose of generating an attitude estimate, typically represented as a quaternion [11]. The fundamental principle underlying star trackers is grounded in the fact that stars are extensively cataloged and exhibit relatively consistent and predictable motion, thereby preserving discernible patterns among neighboring stars [16] [17]. Through the identification of these patterns and their subsequent comparison with a preexisting star catalog, star trackers are able to derive accurate attitude data. The star tracker measurement process can be broadly separated into four sequential steps: image capture, centroiding, identification, and attitude estimation. Figure 3.1, adapted from Li et al. (2022) [18], provides a visual representation of this process, which will be further examined in the subsequent discussion.

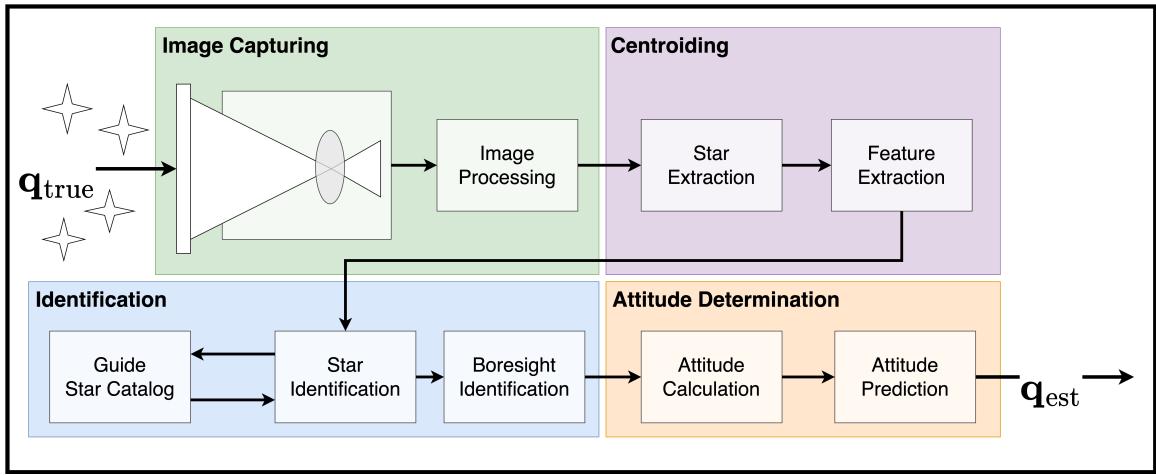


Figure 3.1: High-level measurement process of a star tracker

3.1 Star Catalog

The stars captured by a star tracker are situated on the celestial sphere, which serves as a representation of the stars observed in the sky, modeled as a sphere. These stars are meticulously documented by various organizations and made available in the form of comprehensive star catalogs. Prominent examples include the Yale Bright Star Catalog (BSC5) from Yale University, the Hipparcos Catalog from the European Space Agency, and the Tycho-2 Catalog, also from the European Space Agency. While each catalog may exhibit variations in formatting and specific properties attributed to each star, a subset of fundamental properties remains consistent across all catalogs, including the star ID, right ascension, declination, (typically apparent) magnitude, and proper motion [16] [17]. These properties provide crucial identification information, including positional, motion, and brightness data for each star. Despite the potential presence of measurement errors within the catalog, which may be exacerbated by problems related to timestamping to account for the motion of stars, the catalog can still be reliably considered as a source of accurate data. This is based on the premise that any errors external to the star tracker would consistently affect both the ideal and the modelled star tracker. A representative excerpt from the Yale Bright Star Catalog is presented in Table 3.1[17].

In the celestial coordinate system, the star's position on the celestial sphere is denoted by two spherical coordinates: right ascension (α) and declination (δ). These coordinates are defined within the Earth-Centered Inertial (ECI) frame and facilitate the determination of the star's vector. Specifically, right ascension represents the angular measurement along the celestial equator from a fixed reference point (e.g., the First Point of Aries), while declination indicates the angular measurement from the celestial equator along the meridian of the celestial sphere. The vector defining the position of a star on the celestial sphere in the ECI frame is given by Equation

3.1. The \square_{\oplus} subscript denotes representation in the ECI frame.

$$\vec{V}_{\oplus} = \begin{bmatrix} \cos \alpha \cos \delta \\ \sin \alpha \cos \delta \\ \sin \delta \end{bmatrix} \quad (3.1)$$

Catalog Number	B1950 Right Ascension	B1950 Declination	Spectral Type	V Magn. x100	R.A. Proper Motion	Dec. Proper Motion
...						
123	0.138637	0.951592	B8	4.73	2.181662e-07	-4.848137e-08
124	0.138259	0.922222	K2	5.60	-2.569512e-07	-9.211460e-08
125	0.137081	-0.851784	A0	4.77	6.932835e-07	8.241832e-08
...						

Table 3.1: Sample star catalog entry from Yale Bright Star Catalog

It is important to note that the acceptable range of values for the right ascension lies between 0 and 2π radians, while the range for declination spans from $-\pi/2$ to $\pi/2$ radians. Any value falling outside these prescribed intervals can be rectified by applying an additional rotation of π radians to the right ascension. To provide visual context, Figure 3.2 showcases a segment of the celestial sphere as observed by a star tracker and the distribution of stars when viewed from a flattened perspective.

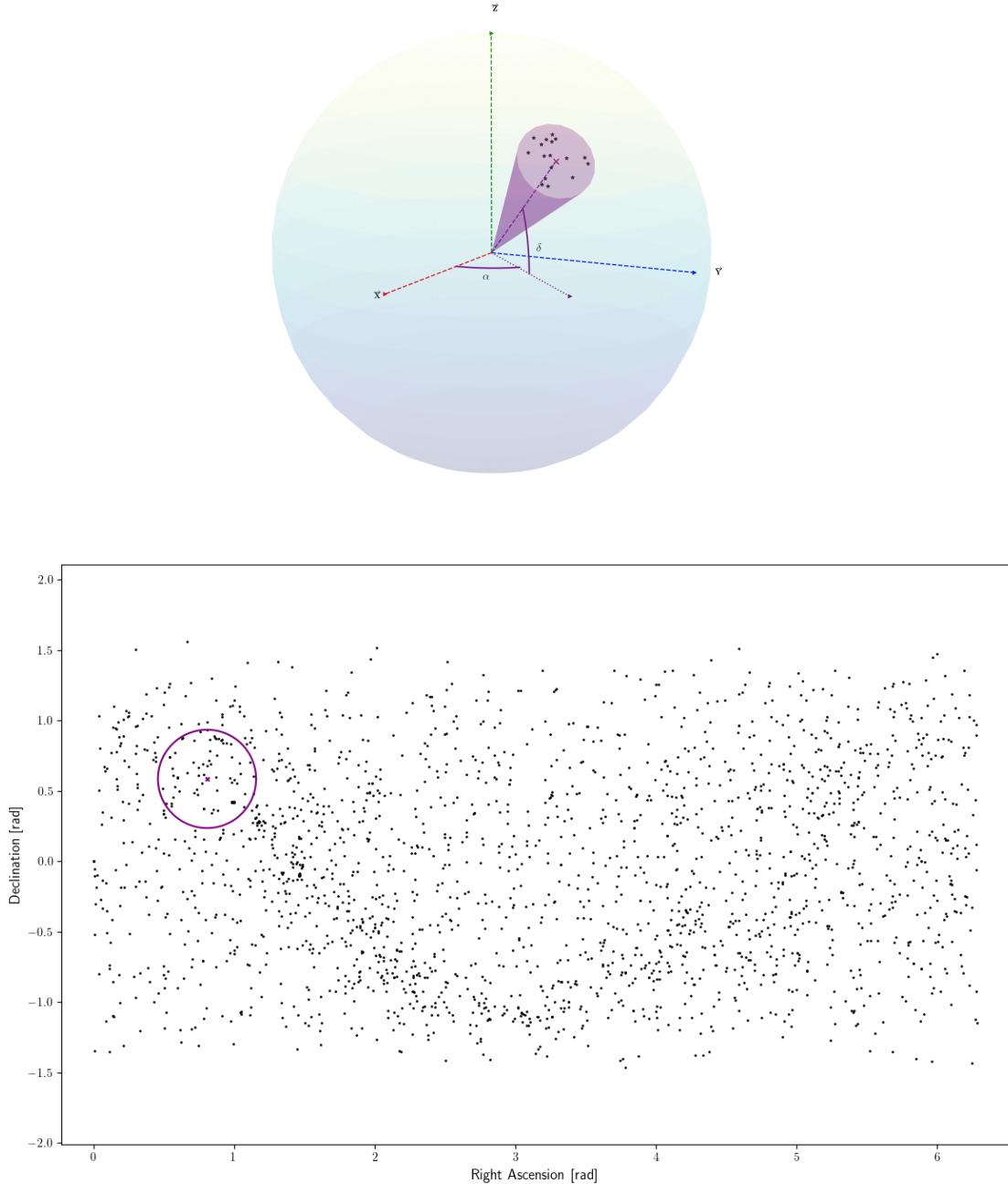


Figure 3.2: Celestial sphere (above) and Plane (below) filtered from the Yale Bright Star Catalog (BSC5) on stars with apparent magnitude 5.2 and less; $(\alpha, \delta) \equiv (46.25, 33.54)^\circ$

Based on the direction of the boresight of the star tracker, denoted by the angles $[\alpha, \delta]^T$ symbolizing the right ascension and declination, respectively, the star catalog can be selectively filtered to generate feasible projections or simulations that illustrate

what a star tracker might observe. The projection can be used to generate simulated images for star tracker testing. This process determines which stars fall within the field of view (FOV) of the star tracker and establishes their precise positions within the ECI frame. The process presented in Algorithm 1 provides a comprehensive description of the procedure employed to determine the list of stars and their ECI representations¹.

Algorithm 1 Process for Star Projection Determination

```

 $\Lambda$ : Star Catalog
 $\alpha$ : Right Ascension
 $\delta$ : Declination
FOV: Field of View

function ANGLESТОVECTOR( $\alpha, \delta$ )
    return  $[\cos \alpha \cdot \cos \delta, \sin \alpha \cdot \cos \delta, \sin \delta]^T$ 
end function

function STARPROJECTION( $\Lambda, \alpha, \delta, \text{FOV}$ )
    Set boresight
     $\vec{B} = \text{AnglesToVector}(\alpha, \delta)$ 

    Filter Star Catalog on  $\alpha, \delta$ 
     $\Lambda_\alpha = \{x \text{ in } \Lambda \text{ if } (\alpha - \text{FOV}/2) \leq x_\alpha \leq (\alpha + \text{FOV}/2)\}$ 
     $\Lambda_\delta = \{x \text{ in } \Lambda \text{ if } (\delta - \text{FOV}/2) \leq x_\delta \leq (\delta + \text{FOV}/2)\}$ 
     $\Lambda_B \leftarrow \Lambda_\alpha \cap \Lambda_\delta$   $\triangleright$  Visible stars is in the intersection of  $\Lambda_\alpha$  and  $\Lambda_\delta$ 

    Set ECI Vectors
     $\vec{V}_\oplus = \{\text{AnglesToVector}(x) \text{ for } x \in \Lambda_B\}$   $\triangleright$  Set ECI Vectors
     $\vec{V}_\oplus = \{x \text{ in } \vec{V}_\oplus \text{ if } \arccos(x \cdot \vec{B}) \leq \text{FOV}/2\}$   $\triangleright$  Remove stars outside FOV
     $\Lambda_B \leftarrow \vec{V}_\oplus$   $\triangleright$  Store ECI Vector in star list
    return  $\Lambda_B$ 
end function

```

¹The provided algorithm returns a set of stars in a rectangular FOV about the boresight as opposed to a more traditional circular FOV. This algorithm is used to quickly filter the star catalog and drastically reduce the search space; an optional step of filtering on the angle formed by the star and the boresight with half the FOV can also be taken to remove any stars that reside outside the circular FOV but within the rectangular one.

An optional intermediate step can be taken during the `StarProjection` process to filter out stars in the catalog by their magnitude based on the maximum magnitude a star tracker can register given its quantum efficiency and sensitivity of its sensor; this is further discussed in Section 4.2 discussing sensor noise.

3.2 Hardware Interfaces

During the stage of image capture, the star tracker operates by capturing an image of the celestial sphere. The star tracker hardware can be effectively represented using the pinhole projection model, which serves as a simplified version of an optical sensor, encompassing the fundamental components of a camera [19]. Optical sensors operate by opening an aperture that enables the passage of light into the system. The light is focused by a lens and directed towards the image plane, a photosensitive sheet comprised of pixels, a physical interface with the capability of being electrically excited via photon collision. Within the pinhole projection model, the camera is depicted through two principal elements: the focal point and the image plane. The focal point signifies the location where light enters the system and maintains a specific distance from the center of the image plane. A visual representation of the pinhole projection model and its key components is presented in Figure 3.3². This model simplifies the camera and assists in determining the precise position on the image plane where an incoming signal will synthesize. To appear on the image plane, the signal must originate from within the camera’s designated FOV, which is a function of the focal length of the camera and the size of the image plane. The FOV of the sensor can be calculated by the following expression:

$$FOV = \arctan\left(\frac{w}{2F}\right)$$

²Reflections of the image about the Focal Point are an artifact of the simplification and are not representative of the subject.

where w represents the image width in pixels and F represents the focal length in pixels³.

During image capture, the physical signal is transformed into a digital signal through the measurement of voltage within each pixel well, which is a result of photon collision-induced excitation. Consequently, the translation of photons into a brightness reading introduces inherent noise into the resulting image. In this context, noise is characterized as the presence of excitation within the pixels even in the absence of an actual signal, such as that originating from a star. Failure to account for this noise can lead to the misinterpretation of a spot as a star by the centroiding algorithm, thereby resulting in erroneous identification of the genuine stars depicted in the image. Further exploration of the impact of noise during the measurement process can be found in Section 4.2.2.

3.3 Centroiding

The centroiding phase directly proceeds the image capturing phase and aims to find the center of each captured star and form features based on vectors formed from the boresight of the star tracker to the star in the image. During nominal star tracker operations, the only observable information to the sensor is the star position on the image plane. By employing vector notation, if the object's position is known and it must pass through the focal point, it becomes possible to calculate the vector between the object and its corresponding image. It can be seen that the vector, \vec{V}_* , between the object and its image on the focal array is given by Equation 3.2 where x and y are the X and Y distances, respectively, from the center of the image plane in units of pixels and F is the focal length of the sensor in units of pixels. The vector \vec{V}_*

³The Focal Length is typically given in units of millimeters but can be converted to units of pixels using the size of each pixel on the focal array, usually given in μm -by- μm per pixel from the focal array datasheet.

is often denoted as the camera vector and is the representation of the star's position in the body-fixed frame. Note that the \square_\star subscript indicates a vector in the body-fixed reference frame. The rotation between a star's position in the ECI frame and representation in the body-fixed frame represents the attitude of the star tracker with respect to the ECI frame.

$$\vec{V}_\star = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = \frac{1}{\sqrt{x^2 + y^2 + F^2}} \begin{bmatrix} -x \\ -y \\ F \end{bmatrix} \quad (3.2)$$

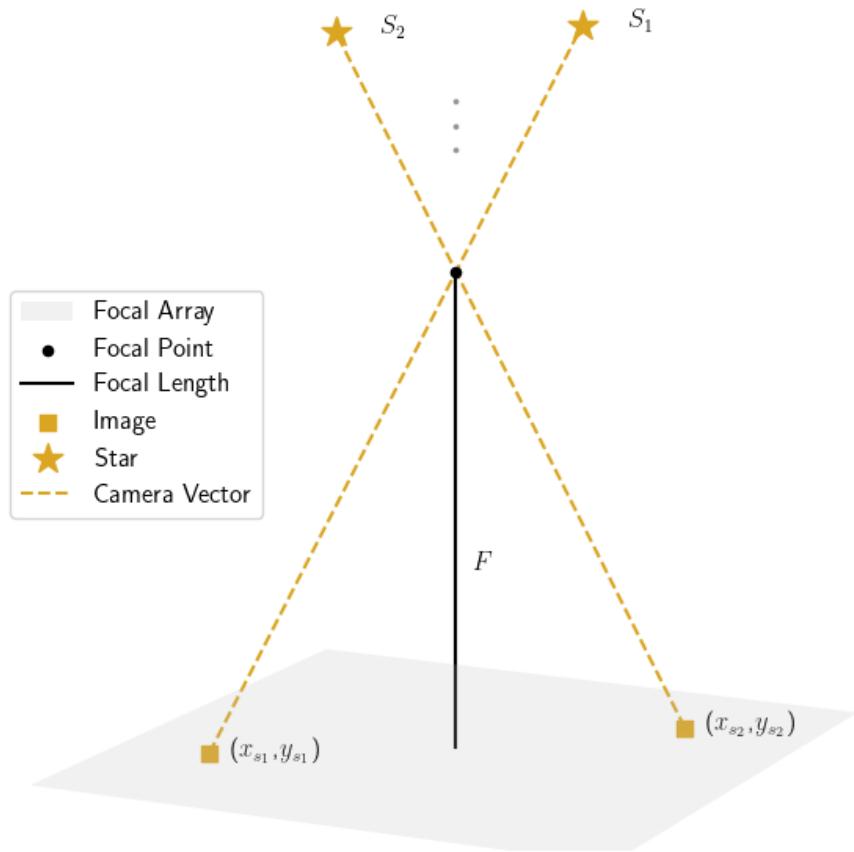


Figure 3.3: Pinhole projection model

From a computational perspective, an image can be perceived as a collection of values that represent pixel data, often denoting brightness or color values. To identify regions of interest, namely areas where stars are anticipated to be present, a star tracker employs various techniques. One commonly utilized approach is the Connected-Component Algorithm, which groups objects together based on predefined

criteria and assigns labels to them. In the specific context of star centroiding, the Connected Component Algorithm assigns a label to a pixel if it satisfies a particular criterion, such as surpassing a predetermined threshold of brightness.

The algorithm operates by examining each pixel within the image. Upon meeting the specified condition, the algorithm proceeds to inspect the immediate neighboring pixels to determine if they already possess a label. If a label is found, the same label is employed to identify the current pixel. Conversely, if no label exists, a new label is generated specifically for that pixel. This process iterates for the entire image. For an image containing n stars, the algorithm generates n labels, each corresponding to a distinct set of associated pixels. These sets of pixels delineate the likely locations of stars within the image, effectively representing the regions of interest associated with each star. Algorithm 6 in Appendix B illustrates the procedural steps of the Connected Component Algorithm for ascertaining the regions of interest relevant to a star tracker.

Once the regions of interest have been determined for each star, several techniques can be employed to calculate the star centroid. Among these techniques, the center of mass approximation is commonly utilized due to its simplicity and reasonable accuracy. Equations 3.3-3.5 outline the center of mass approximation, where I represents the intensity or brightness value of a specific pixel, DN denotes the total brightness within the ROI, and \bar{x} and \bar{y} represent the centroid's x and y coordinates, respectively. For each region of interest, this method is applied, yielding a unique centroid, (\bar{x}, \bar{y}) , for that particular region [11].

$$DN = \sum_{\substack{x=\text{ROI}, \text{ end} \\ x=\text{ROI}, \text{ Start}}} \sum_{\substack{y=\text{ROI}, \text{ end} \\ y=\text{ROI}, \text{ Start}}} I_{x,y} \quad (3.3)$$

$$\bar{x} = \sum_{\substack{x=\text{ROI}, \text{ end} \\ x=\text{ROI}, \text{ Start}}} \sum_{\substack{y=\text{ROI}, \text{ end} \\ y=\text{ROI}, \text{ Start}}} \frac{x \cdot I_{x,y}}{DN} \quad (3.4)$$

$$\bar{y} = \sum_{x=\text{ROI}, \text{ Start}}^{x=\text{ROI}, \text{ end}} \sum_{y=\text{ROI}, \text{ Start}}^{y=\text{ROI}, \text{ end}} \frac{y \cdot I_{x,y}}{DN} \quad (3.5)$$

The accuracy of the center of mass approach is a function of the intensity of the star and the spread of the star signal as presented by Liebe, typically on the order of 0.1 pixels [11].

3.4 Star Identification

Features play a vital role in the process of star identification within images and are derived from the centroids determined in the previous stage. These features, specific to each implementation of identification, serve as metrics that establish connections between stars and are utilized for comparison with the guide star catalog. A range of commonly used features includes interstar angles, which are defined as the angles formed by the vectors connecting pairs of stars, ratios of triangle legs formed by 3 stars, or the radial distance of the star from the center of the image. These features are then compared against a pre-compiled catalog known as the guide star catalog. The guide star catalog contains information about each unique pattern and the corresponding star identification numbers. It includes every combination of stars within the star catalog provided the specific set of stars can be observed within the same image and the maximum magnitude of the star is visible by the sensor. This approach reduces the likelihood of erroneous star identification and reduces the overall search space for the identification algorithm.

By utilizing the features obtained, the identification method is able to establish a mapping between each star in the image and a corresponding star in the true catalog, where the star's position in the ECI frame is known. One increasingly favored technique for star identification in star trackers is the Pyramid Algorithm. This algorithm, developed by Mortari et al. (2004), demonstrates robustness against false

stars and achieves near-certain identification when at least four stars are visible in the image. Based on the k-Vector method, the Pyramid Algorithm forms pyramids of four stars and utilizes interstar angles to generate features within the image and employs a binary search technique to precisely determine the stars involved in each angle, mapping them against a one-dimensional list. It is worth noting that the Pyramid Algorithm expands upon the k-Vector technique by requiring a minimum of four stars to form a feature, resulting in enhanced robustness and reduced computation time by requiring additional thresholds to be met [20]. According to reports, identification can be achieved within a fraction of a second without any prior information [20].

3.5 Operating Modes

The star tracker, in addition to its main process phases, has 2 operating modes. If a star tracker has no *a priori* knowledge of the spacecraft attitude or is relatively uncertain in its measurement, then the star tracker will map the captured stars against the entire guide star catalog. This is called the “Lost-in-Space” problem and is typically resource-intensive and requires a longer time before an attitude can be acquired. If, however, the previous attitude is well-known or recent, the star tracker, using other onboard sensors such as gyroscopes, can estimate its attitude and only consider a small subset of the guide star catalog near the estimated attitude. This process is called “Recursive Identification” and can acquire an attitude estimate much quicker than the Lost in Space mode, although it requires some recent and confident attitude knowledge.

3.6 Space Environment

The space environment can pose significant challenges to CubeSats and their onboard instruments especially star trackers. It exposes them to extreme and rapidly chang-

ing thermal conditions, varying levels of radiation, micrometeoroids, and the vacuum of space. These factors can cause thermal distortion, radiation damage, physical degradation, and increased sensor noise, potentially impacting the reliability and performance of systems like star trackers. Given these factors, it is crucial to consider the space environment when developing CubeSat designs, including considerations for shielding, thermal control, and use of space-rated materials, to ensure their longevity and the accuracy of the onboard instruments. Comprehensive testing and validation processes are also vital to ensure the spacecraft can withstand these conditions throughout its mission lifecycle. In their respective studies, Jia et al. (2010) and Eisenman et al. (1998) highlight thermal drift and sensor noise as principal sources of error in star tracker operations. These errors are predominantly induced by the thermal and radiation conditions prevalent in Low Earth Orbit (LEO).

3.6.1 Thermal Environment

The thermal environment in Low Earth Orbit (LEO) is a highly dynamic and extreme setting that can significantly impact spacecraft. This environment is characterized by significant fluctuations in temperature due to rapid transitions between intense solar heating and deep-space cooling during each orbit cycle. These transitions result from the spacecraft's exposure to the Sun's radiation when in sunlight and exposure to the cold, nearly absolute zero temperatures of space during the eclipse phase of the orbit. The high thermal variance requires spacecraft to be designed with sophisticated thermal control systems to ensure consistent operational temperature for their sensitive components and instruments. Unchecked temperature gradients can lead to thermal stress, distortions, and potential failure of onboard systems, affecting the overall performance, reliability, and lifespan of the spacecraft and its onboard sensors.

In Garzón et al. (2018), a 3U CubeSat in LEO was determined to experience oscillatory temperature fluctuations between 20° C and 40°C, but could experience temperatures as cold as 0°C and as high as 60°C depending on the satellite position and time of year. Fluctuations in temperature can also impact the electronic components of the star tracker. For instance, the sensitivity and noise of sensor may vary with temperature, affecting the overall image quality. Depending on the construction of the star tracker, these distortions could additionally lead to misalignment of optical elements or changes in focal length, resulting in blurred or shifted star images.

3.6.2 Radiation Environment

The radiation environment is another concern for CubeSats in LEO. In the space environment, radiation refers to the emission and propagation of energetic particles and electromagnetic waves. These particles can originate from various sources, including the Sun, galactic cosmic rays, and trapped radiation belts around celestial bodies. Spacecraft are exposed to this radiation in orbit which can have detrimental effects on their components and systems. High-energy particles can penetrate spacecraft materials, leading to electronic failures, degradation of solar cells, and damage to sensitive instruments and can be measured using the Gray or Rad, a unit to measure the absorbed dose of radiation. While radiation can affect the electronics of the star tracker, resulting in a non-deterministic effect on performance, a measurable concern is the interaction between radiation and the optical sensor. Radiation can have detrimental effects on optical sensors, resulting in increased overall noise levels and the generation of bright spots on images that simulate false stars. The presence of radiation can interfere with the sensor's ability to accurately capture and process light, leading to a higher level of background noise in the images. Additionally, radiation-induced pixel defects can create hot or bright spots, resembling stars, which are not actually present in the observed scene. These effects can compromise

the quality and reliability of the centroiding and identification phases during the star tracker measurement process.

For lower inclinations (e.g., less than 28°), it is expected that spacecraft will absorb approximately 100-1,000 rad(Si) per year of operation. In contrast, higher inclinations, which CubeSats are more likely to reside in [21], are expected to experience a higher absorption rate of 1,000 - 10,000 rad(Si) per year [22]. In the experiment performed by Hopkinson specifically investigating the effects of radiation on an optical sensor meant for a star tracker, it was seen that the main effects were the increase in dark current and decrease in responsivity [23]. Radiation directly contributes to total dark current experienced by a sensor. Dark current is caused by the excitation of the pixels in the absence of an incoming signal [24]; in the case of radiation, the pixels are rightfully excited, however, not from the intended subject of the image. Charged particles colliding with the sensor is inherently a stochastic process and, as such, there is no systematic method for removing or differentiating signal caused by radiation or an incoming signal. In a separate experiment analyzing the effects of radiation in LEO on optical sensors, Rushton confirms the environment can worsen conditions for dark current and fixed pattern noise. In the experiment performed by Rushon, a set of optical sensors were irradiated with similar levels of radiation found in LEO and determined that the average dark current had risen 100% - 150% and the fixed pattern noise had risen to 200% of its original value [25].

Understanding the underlying measurement process is integral to developing a faithful model examining error propagation in the star tracker. By replicating the overall process, errors can be intelligently tracked through the system and comparisons between ideal operation and true operation can be accurately described.

Chapter 4

MEASUREMENT PROCESS MODEL

4.1 Model Overview

To model the star tracker measurement process and examine the effects of disturbances, it is critical to model errors that are faithful to the underlying principles. Error propagation is defined as “a basic problem analyzing the uncertainty of reliable systems” and, according to Jia et al. (2010), “[t]he most important factors that affect star tracker accuracy include thermal drift, optical aberration, detector noise, and systematic error of star image centroid estimation algorithm” [26]. Read Eisenman and Liebe also present an error tree for star trackers where the main concerns for star tracker accuracy include thermal drift, optic errors, centroiding errors, sensor noise, and algorithmic errors [27]. It is clear that a successful star tracker measurement model should include considerations for these factors at a minimum.

In order to assess the accuracy of a parameterized star tracker, the Monte Carlo technique was utilized. This involved the selection of a series of input parameters, denoted as $\mathbf{X} \in \mathbf{R}^n$, from separate probability density functions $[g_1(\xi), g_2(\xi) \dots g_n(\xi)]^T$, each representing a distribution of potential states. These parameters were then incorporated into an objective function, $\Gamma(\mathbf{X})$, which serves to quantify the discrepancy between the estimated and true attitudes. The aforementioned process was reiterated multiple times, and in accordance with the Law of Large Numbers, it was anticipated that the mean of the outcomes from a vast quantity of trials should converge towards the expected value. This convergence becomes more pronounced with the increasing number of trials [28], thereby facilitating an accurate portrayal of the expected performance of the specified star tracker. The star tracker measurement process

model, as proposed in this study, has been derived from first principles, with errors being simulated at each step of the process. The model thoroughly investigates the impacts stemming from hardware aberrations (i.e., deviations from the pinhole projection model), sensor noise, and centroiding performance on the attitude estimate. Lens distortion, however, is not necessarily considered in this model. Defined as the deviation of the light rays once passed through the lens, there exist several calibration techniques to reduce or eliminate lens distortion [29][30] and the purpose of this model is to investigate lesser known disturbances. These factors were chosen for two primary reasons: firstly, their explicit reference by Eisenman et al. (1998), and Jia et al. (2010), and secondly, the fact that any other source of error can be translated in terms of its influence on these explicit factors. For example, the thermal environment in LEO can be analyzed in terms of its effects on variations in focal length and its influence on sensor noise; this is further examined in Chapter 4.3.2. Table 4.1 illustrates the set of input parameters along with the distribution characterizing them. In conjunction with the input parameters, each simulation is also assigned a random attitude represented by a uniformly distributed right ascension, declination, and roll angle. Every state is subsequently evaluated based on the error in attitude estimation relative to the true attitude. This process is repeated numerous times to sufficiently encapsulate the effects of each input on attitude determination. The method for generating and assessing each star tracker state is presented in Algorithm 2.

In light of the extensive range of input distributions and the theoretical coverage area, it is impractical to conduct an arbitrary large number of trials, for instance, for a confidence interval. Instead, a stopping criterion could be adopted to generate data that has achieved statistical convergence after being processed through the objective function [31]. The model operates in groups of 100 simulations, designated as δ , which are subsequently appended to a comprehensive dataset, labeled as Δ , encompassing

Algorithm 2 Process of a Simulation Run

```

function COMPUTESTARTRACKERACCURACY

    
$$\mathbf{X} \leftarrow \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_n \end{bmatrix} \in \begin{bmatrix} g_1(\xi) \\ g_2(\xi) \\ \dots \\ g_n(\xi) \end{bmatrix}$$

     $\alpha \sim U(0, 2\pi)$ 
     $\delta \sim U(-\pi/2, \pi/2)$ 
     $\phi \sim U(0, 2\pi)$             $\triangleright$  Set random right ascension, declination, and roll angle
     $\vartheta \leftarrow \Gamma(\mathbf{X}, \alpha, \delta, \phi)$            $\triangleright$  Evaluate star tracker state
    return  $\vartheta$ 
end function

```

each run from each batch. Following each batch, the standard deviation of the computed accuracy from the extensive dataset is determined, and once the percentage change in standard deviation meets a predetermined tolerance, for instance, 1e-5, the simulation ceases. In the context of this model, the accuracy of the star tracker is given by the standard deviation of the error. Ceasing the simulation once the change in standard deviation with subsequent batches falls below the tolerance of 1e-5 is equivalent to stating the precision of the current data is within 0.001% of the total error. If a tighter tolerance level is set, it will demand a larger dataset, but it will also deliver more accurate results. Conversely, with a more lenient tolerance level, it will necessitate less data; however, it is important to note that this would result in less precision.

Since the output of the $\Gamma(\mathbf{X})$ function denotes the magnitude of deviation between the true and estimated rotation, meaning the output is positive semi-definite, the ensuing distribution of values adheres to a half-normal distribution. Half-normal distributions are characterized by having a zero-mean and a variance proportional to the underlying normal distribution, that is,

$$\{\vartheta\} = |\mathcal{N}(0, \sigma_{\text{half}}^2)| \quad (4.1)$$

where $\{\vartheta\}$ represents the set of values computed from the simulation.

Expecting a half-normal distribution is particularly valid for the model as it exclusively generates error data. The star tracker model returns the error between the true and estimated rotation, however, the direction of the error is assumed to be uniformly around the target over several simulations. The half-normal distribution also maintains some of the desirable properties of the normal distribution allowing for effective analysis and interpretation of the error data generated by the sensor model.

Algorithm 3 presents the overarching simulation architecture.

Algorithm 3 Simulation Architecture

```

 $\sigma_{\text{Ratio}} = 1$ 
 $\sigma_{\text{Previous}} = 1$        $\triangleright$  These are arbitrary values and are used to enter the while loop
 $\Delta = \emptyset$                  $\triangleright$  Initialize the large dataset as an empty set

Compute sample dataset
while  $\sigma_{\text{Ratio}} > 1e-5$  do
     $\delta = \emptyset$                        $\triangleright$  Reset batch dataset
    for 100 simulations do
         $\vartheta = \text{ComputeStarTrackerAccuracy}$ 
         $\delta = \delta \cup \vartheta$             $\triangleright$  Compute 100 simulations
    end for
     $\Delta = \Delta \cup \delta$              $\triangleright$  Append batch dataset to the sample dataset
     $\sigma_{\text{New}} = \sigma_{MC}(\Delta)$ 
     $\sigma_{\text{Ratio}} = |\sigma_{\text{Previous}} - \sigma_{\text{New}}| / \sigma_{\text{Previous}}$        $\triangleright$  Recompute the stopping criteria
     $\sigma_{\text{Previous}} = \sigma_{\text{New}}$            $\triangleright$  Update the previous standard deviation
end while

 $\sigma^* = \sigma(\Delta)$ 
 $\mu^* = \mu(\Delta)$ 

```

ECI to Body-Fixed Rotation

The star tracker measurement process begins with the analysis of a star image, wherein the body-fixed vectors are initially determined, followed by the identification of corresponding vectors in the Earth-Centered Inertial (ECI) frame. However, in the simulation, the attitude is predetermined, resulting in a reversed operation.

In this case, the `StarProjection` algorithm determines the set of vectors in the ECI frame, which are subsequently rotated into the body-fixed reference frame. This approach enables the star tracker measurement process model to simulate realistic star scenes, as opposed to randomly generating vectors within the field of view.

While Equation 3.2 can be utilized to determine the star vector in the body-fixed reference frame given the image's position, it is necessary to derive a rotation that transforms the set of vectors directly from the ECI frame into the body-fixed frame as the image properties are yet to be defined. It is important to emphasize that the body-fixed frame and the ECI frame are not necessarily aligned. In this particular model, the alignment is defined such that the boresight of the star tracker corresponds to the body-fixed $+Z$ axis, which is aligned with the ECI $+X$ axis. Similarly, the body-fixed $+X$ axis aligns with the ECI $-Z$ axis. Essentially, the body-fixed reference frame is rotated by $\pi/2$ radians about the ECI $+Y$ axis. The intentional misalignment of the reference frames serves the purpose of ensuring consistency with the body-fixed vector representation described in Equation 3.2, where the boresight of the vector is aligned with the $+Z$ axis in the body-fixed frame. Moreover, the boresight of the vector needs to be defined in the Earth-Centered Inertial (ECI) frame based on the attitude of the star tracker. These combined considerations necessitate that the reference frame of the star tracker be deliberately misaligned with that of the ECI reference frame.

To derive the rotation matrix, $\mathbf{C}_{\star\oplus}$, defining the rotation from a vector in the ECI frame, \mathbf{V}_\oplus , to a vector in the body-fixed frame, \mathbf{V}_\star , consider the vectrix form of the reference frame defined in Section 2.3. The rotation matrix $\mathbf{C}_{\star\oplus}$ is defined as the dot product between the reference frames \mathcal{F}_\star and \mathcal{F}_\oplus^T . Both reference frames define the

same physical vector, \vec{V} ; i.e.,

$$\mathcal{F}_* = \begin{bmatrix} \vec{x}_* \\ \vec{y}_* \\ \vec{z}_* \end{bmatrix} ; \quad \mathcal{F}_\oplus = \begin{bmatrix} \vec{x}_\oplus \\ \vec{y}_\oplus \\ \vec{z}_\oplus \end{bmatrix} \quad (4.2)$$

$$\vec{V} = \mathcal{F}_*^T \mathbf{V}_* = \mathcal{F}_\oplus^T \mathbf{V}_\oplus \quad (4.3)$$

$$\mathbf{V}_* = \mathcal{F}_* \cdot \mathcal{F}_\oplus^T \mathbf{V}_\oplus \quad (4.4)$$

$$\mathbf{C}_{*\oplus} = \mathcal{F}_* \cdot \mathcal{F}_\oplus^T \quad (4.5)$$

where the rotation matrix $\mathbf{C}_{*\oplus}$ is defined as

$$\mathbf{C}_{*\oplus} = \begin{bmatrix} \vec{x}_{*\oplus} \\ \vec{y}_{*\oplus} \\ \vec{z}_{*\oplus} \end{bmatrix} \quad (4.6)$$

$$\vec{x}_{*\oplus}, \vec{y}_{*\oplus}, \vec{z}_{*\oplus} \in \mathbf{R}^{1 \times 3} \quad (4.7)$$

In the case of the boresight, the vector is defined in the ECI frame as given by Equation 3.1 and the corresponding vector in the body-fixed frame is defined as the +Z axis, or $[0, 0, 1]^T$. These can be substituted in Equation 4.4; i.e.,

$$\mathbf{V}_\oplus = [\cos \alpha \cos \delta, \sin \alpha \cos \delta, \sin \delta]^T \quad (4.8)$$

$$\mathbf{V}_* = [0, 0, 1]^T \quad (4.9)$$

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{C}_{*\oplus} \cdot \begin{bmatrix} \cos \alpha \cos \delta \\ \sin \alpha \cos \delta \\ \sin \delta \end{bmatrix} \quad (4.10)$$

If Equation 4.10 holds true for any α and δ which is the case for the model, it can be shown that \vec{z}_* must also be equivalent to $[\cos \alpha \cos \delta, \sin \alpha \cos \delta, \sin \delta]^T$. Consider

the inverse of Equation 4.10; i.e.,

$$\begin{bmatrix} \cos \alpha \cos \delta \\ \sin \alpha \cos \delta \\ \sin \delta \end{bmatrix} = \mathbf{C}_{\star\oplus}^T \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.11)$$

Equation 4.11 can only hold true if and only if the third column of $\mathbf{C}_{\star\oplus}^T$ is equivalent to $[\cos \alpha \cos \delta, \sin \alpha \cos \delta, \sin \delta]^T$. This can be verified by checking against the initial form given in Equation 4.10; i.e., let $\vec{\mathbf{z}}_{\star\oplus}$ be the third column of $\mathbf{C}_{\star\oplus}^T$ and, equivalently, the third row of $\mathbf{C}_{\star\oplus}$.

$$1 = \vec{\mathbf{z}}_{\star\oplus} \cdot \begin{bmatrix} \cos \alpha \cos \delta \\ \sin \alpha \cos \delta \\ \sin \delta \end{bmatrix} \quad (4.12)$$

$$1 = [\cos \alpha \cos \delta, \sin \alpha \cos \delta, \sin \delta] \cdot \begin{bmatrix} \cos \alpha \cos \delta \\ \sin \alpha \cos \delta \\ \sin \delta \end{bmatrix} \quad (4.13)$$

$$1 = \cos^2 \alpha \cos^2 \delta + \sin^2 \alpha \cos^2 \delta + \sin^2 \delta \quad (4.14)$$

$$1 = \cos^2 \delta (\cos^2 \alpha + \sin^2 \alpha) + \sin^2 \delta \quad (4.15)$$

$$1 = \cos^2 \delta + \sin^2 \delta \quad (4.16)$$

$$1 = 1 \text{ QED} \quad (4.17)$$

Furthermore, based on the order of rotations, it is defined that the $+X$ axis in the body-fixed reference frame is always exactly $-\pi/2$ radians in the declination relative to the boresight, i.e.,

$$\vec{\mathbf{x}}_{\star\oplus} = [\cos(\alpha) \cos(\delta - \pi/2), \sin(\alpha) \cos(\delta - \pi/2), \sin(\delta - \pi/2)] \quad (4.18)$$

$$= \begin{bmatrix} \cos \alpha \sin \delta, \sin \alpha \sin \delta, -\cos \delta \end{bmatrix} \quad (4.19)$$

The final axis completes the dextral orthogonal basis and can be computed by the cross product of the +Z and +X axes; i.e.,

$$\vec{\mathbf{y}}_{\star\oplus} = \vec{\mathbf{z}}_{\star\oplus} \times \vec{\mathbf{x}}_{\star\oplus} \quad (4.20)$$

$$\vec{\mathbf{y}}_{\star\oplus} = \begin{bmatrix} \cos \alpha \cos \delta \\ \sin \alpha \cos \delta \\ \sin \delta \end{bmatrix}^T \times \begin{bmatrix} \cos \alpha \sin \delta \\ \sin \alpha \sin \delta \\ -\cos \delta \end{bmatrix}^T \quad (4.21)$$

$$\vec{\mathbf{y}}_{\star\oplus} = \begin{bmatrix} -\sin \alpha \cos^2 \delta - \sin \alpha \sin^2 \delta \\ \cos \alpha \sin^2 \delta + \cos \alpha \cos^2 \delta \\ \cos \alpha \cos \delta \sin \alpha \sin \delta - \sin \alpha \cos \delta \cos \alpha \sin \delta \end{bmatrix}^T \quad (4.22)$$

$$= \begin{bmatrix} -\sin \alpha, \cos \alpha, 0 \end{bmatrix} \quad (4.23)$$

The matrix formed by $[\vec{\mathbf{x}}_{\star\oplus}, \vec{\mathbf{y}}_{\star\oplus}, \vec{\mathbf{z}}_{\star\oplus}]^T$ only considers the right ascension and declination components of the true attitude. To consider the roll about the boresight, the \mathbf{R}_z rotation must be prepended; i.e,

$$\mathbf{C}_{\star\oplus} = \mathbf{R}_z(\phi) \begin{bmatrix} \vec{\mathbf{x}}_{\star\oplus} \\ \vec{\mathbf{y}}_{\star\oplus} \\ \vec{\mathbf{z}}_{\star\oplus} \end{bmatrix} \quad (4.24)$$

The final rotation matrix fully rotating the vectors from the ECI frame into the body-fixed frame is presented in Equation 4.26.

$$\mathbf{C}_{\star\oplus} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\alpha) \sin(\delta) & \sin(\alpha) \sin(\delta) & -\cos(\delta) \\ -\sin \alpha & \cos \alpha & 0 \\ \cos \alpha \cos \delta & \sin \alpha \cos \delta & \sin \delta \end{bmatrix} \quad (4.25)$$

which can be simplified to the following expression:

$$\mathbf{C}_{\star\oplus} = \mathbf{R}_z(\phi)\mathbf{R}_y(\delta - \pi/2)\mathbf{R}_z(-\alpha) \quad (4.26)$$

The rotation matrix $\mathbf{C}_{\star\oplus}$ will be updated and used for each simulation to transform the set of ECI vectors derived from the `StarProjection` algorithm to prepare the set of body-fixed vectors to undergo distortion to analyze error propagation in the star tracker measurement process.

4.2 Error Modeling

Errors in the measurement process model can be characterized by analyzing the physics behind their presence and how its intensity affects the final estimate. Figure 4.1 modifies Figure 3.1 to illustrate where different error sources originate during the measurement process. It is imperative to analyze each effect in the order of which they appear in the measurement process to examine how they may compound over the length of the process.

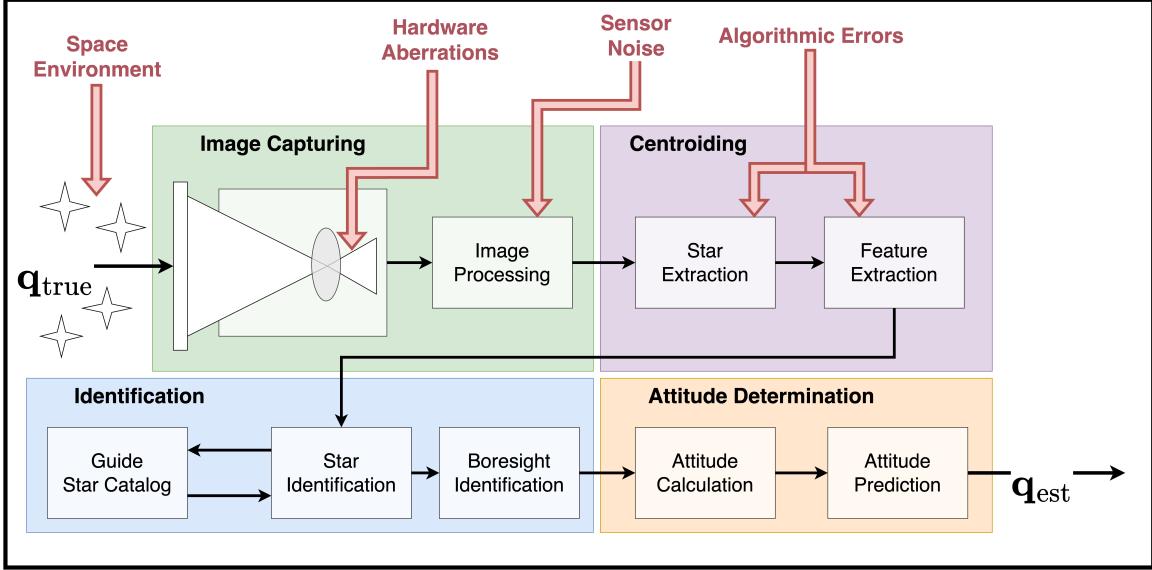


Figure 4.1: Error sources and propagation within the star tracker measurement process

4.2.1 Hardware Analysis

The physical hardware is the first interaction of the star tracker with its environment and, as such, will always be the first point of error. Star trackers and other optical sensors can be modeled with the pinhole projection model, as presented in Figure 3.3. However, translation and rotation in the image plane and lens distortion each modify the measured camera vector from its true nature.

To analyze the effects of hardware aberration, a model, $\Phi(\mathbf{S})$, was developed to exactly determine the measured camera vector, $\hat{\mathbf{S}}$, as a function of the input, or true, camera vector in the full 6 DOF case. Inputs to the $\Phi(\mathbf{S})$ model include the transformations from the distorted image plane, denoted as the π plane with reference frame $\vec{\mathcal{F}}_\pi$, relative to the ideal image plane, denoted as the γ plane with reference frame $\vec{\mathcal{F}}_\gamma$; e.g.,

$$\hat{\mathbf{S}} = \Phi(\mathbf{S}) \equiv \Phi(\mathbf{S}, \epsilon, \mathbf{C}) \quad (4.27)$$

$$\hat{\mathbf{S}} = \Phi(\mathbf{S}, \epsilon_x, \epsilon_y, \epsilon_z, \phi_x, \theta_y, \psi_z) \quad (4.28)$$

where $\boldsymbol{\epsilon} = [\epsilon_x, \epsilon_y, \epsilon_z]^T$ represent the translation from $\vec{\mathcal{F}}_\gamma$ to $\vec{\mathcal{F}}_\pi$ in the $\vec{\mathcal{F}}_\gamma$ system and $\mathbf{C} = [\phi_x, \theta_y, \psi_z]^T$ represent the rotation of $\vec{\mathcal{F}}_\pi$ with respect to $\vec{\mathcal{F}}_\gamma$ in the $\vec{\mathcal{F}}_\gamma$ system. An uninterrupted derivation of the $\Phi(\mathbf{S})$ model is presented in Appendix A.

Hardware Aberration Model

Suppose there exist two image planes, γ and π , with reference frames $\vec{\mathcal{F}}_\gamma$ and $\vec{\mathcal{F}}_\pi$ located at the center of their respective planes and some star, S , beyond the focal point. The goal of the Hardware Aberration Model is to pose a relation of S in the $\vec{\mathcal{F}}_\pi$ system where the system as defined in Figure 3.3 and Equation 3.2 represent the $\vec{\mathcal{F}}_\gamma$ system.

Let the $\square_{a/b,c}$ notation denote a vector from b to a in the reference frame of c . The location of the focal point, F , with respect to γ and π can be written as

$$\mathbf{r}_{F/\gamma,\gamma} = \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix} \quad \mathbf{r}_{\gamma/\pi,\gamma} = \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \end{bmatrix} \quad (4.29)$$

$$\therefore \mathbf{r}_{F/\pi,\gamma} = \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ F + \epsilon_z \end{bmatrix} \quad (4.30)$$

where F also represents the focal length from γ . To rotate the system from $\vec{\mathcal{F}}_\gamma$ to $\vec{\mathcal{F}}_\pi$, a rotation matrix, \mathbf{C} can be derived as

$$\mathbf{C}_{\gamma\pi} = \mathbf{R}_x(\phi_x)\mathbf{R}_y(\theta_y)\mathbf{R}_z(\psi_z) \quad (4.31)$$

such that

$$\mathbf{r}_{F/\pi,\pi} = \mathbf{C}_{\gamma\pi}\mathbf{r}_{F/\pi,\gamma} \quad (4.32)$$

Suppose the star, S , can be represented with by a vector with respect to the focal point; i.e.,

$$\mathbf{S} = \mathbf{r}_{S/F,\gamma} \quad (4.33)$$

where \mathbf{S} represents the true camera vector in the star tracker system. The location of the star can also be written with respect to γ and π such that

$$\mathbf{r}_{S/\gamma,\gamma} = \mathbf{r}_{F/\gamma,\gamma} + \mathbf{r}_{S/F,\gamma} \quad (4.34)$$

and

$$\mathbf{r}_{S/\pi,\gamma} = \mathbf{r}_{\gamma/\pi,\gamma} + \mathbf{r}_{F/\gamma,\gamma} + \mathbf{r}_{S/F,\gamma} \quad (4.35)$$

$$= \mathbf{r}_{F/\pi,\gamma} + \mathbf{r}_{S/F,\gamma} \quad (4.36)$$

Using the knowledge from Equations 4.31 and 4.36, the position of the star can be rewritten in the $\vec{\mathcal{F}}_\pi$ system as

$$\mathbf{C}_{\gamma\pi}(\mathbf{r}_{S/\pi,\gamma}) = \mathbf{C}_{\gamma\pi}(\mathbf{r}_{F/\pi,\gamma} + \mathbf{r}_{S/F,\gamma}) \quad (4.37)$$

$$\mathbf{r}_{S/\pi,\pi} = \mathbf{C}_{\gamma\pi}(\mathbf{r}_{F/\pi,\gamma} + \mathbf{r}_{S/F,\gamma}) \quad (4.38)$$

and finally,

$$\mathbf{r}_{S/\pi,\pi} = \mathbf{r}_{F/\pi,\pi} + \mathbf{C}_{\gamma\pi}\mathbf{r}_{S/F,\gamma} \quad (4.39)$$

Notice that \mathbf{S} has unit length; i.e., the true magnitude of \mathbf{S} has no discernable meaning as the distance to the star from the star tracker is largely irrelevant to the attitude estimation process. Equation 4.39 can be rewritten such that any point, $[x_F, y_F, z_F]^T$, along the \mathbf{S} direction with respect to the focal point can be described by the direction with some magnitude multiplier, λ ; i.e., $[x_F, y_F, z_F]^T = \lambda \mathbf{S}$. With respect to $\vec{\mathcal{F}}_\pi$, any point along \mathbf{S} can be written as

$$\begin{bmatrix} x_\pi \\ y_\pi \\ z_\pi \end{bmatrix} = \mathbf{r}_{F/\pi,\pi} + \lambda \mathbf{C}_{\gamma\pi} \mathbf{r}_{S/F,\gamma} \quad (4.40)$$

where $[x_\pi, y_\pi, z_\pi]^T$ represent any point along the star vector with respect to $\vec{\mathcal{F}}_\pi$ and λ represents some arbitrary multiplier.

The next step is to determine the exact coordinates of where the star vector would intercept with the image plane. By definition, the z_π component of this coordinate is 0 as $\vec{\mathcal{F}}_\pi$ is located on the image plane at its center. This coordinate can be denoted as $[x_\pi^*, y_\pi^*, z_\pi^*]^T$ where

$$\begin{bmatrix} x_\pi^* \\ y_\pi^* \\ z_\pi^* = 0 \end{bmatrix} = \mathbf{r}_{F/\pi,\pi} + \lambda^* \mathbf{C}_{\gamma\pi} \mathbf{r}_{S/F,\gamma} \quad (4.41)$$

The multiplier, λ^* , can be computed by realizing that $z_\pi^* = 0$. Because $\mathbf{r}_{F/\pi,\pi}$, $\mathbf{C}_{\gamma\pi}$, and $\mathbf{r}_{S/F,\gamma}$ are known, λ^* can be exactly determined.

$$\lambda^* = \frac{- (\mathbf{r}_{F/\pi,\pi})_z}{(\mathbf{C}_{\gamma\pi} \mathbf{r}_{S/F,\gamma})_z} \quad (4.42)$$

where the $(\square)_z$ notation represents the z^{th} component of the vector. Notice, the coordinates of the star on the image plane, (x_π^*, y_π^*) , are the only observable informa-

tion to extract star information from. As such, the measured star vector, $\hat{\mathbf{S}}$, can be calculated as shown in Equation 3.2 as if π represented the ideal plane, γ ; i.e.,

$$\hat{\mathbf{S}} = \frac{1}{\sqrt{x_\pi^{*2} + y_\pi^{*2} + F^2}} \begin{bmatrix} -x_\pi^* \\ -y_\pi^* \\ F \end{bmatrix} \quad (4.43)$$

The $\Phi(\mathbf{S})$ model can be applied to each star in a given simulation according to the predetermined distributions to derive the measured star vector in the body-fixed frame. During nominal operations, the vector $\hat{\mathbf{S}}$ is the only information the star tracker is able to determine from its image if the centroid is ideally computed. The $\Phi(\mathbf{S})$ model can be applied to the set of camera vectors, \mathbf{V}_* , that exist within the star tracker FOV for each simulation to compute the measured star vector.

4.2.2 Noise Simulation

Between photons from the star exciting the pixels on the sensor and the sensor reporting the image, noise is generated and distorts the image. Noise in the camera sensor is defined as an unwanted signal perturbing the true signal and is caused by the imperfections in the components making up the sensor. In a star tracker, noise is important to consider [29] as the centroiding step may be unable to determine the star center or may even mistake noise for a star causing errors via misidentification. Noise can appear in the image in two main ways: pixels being partially full and reporting brightness in the absence of a true incident light and the distortion of the true signal on the image plane and is generated in several different methods such as shot noise, dark current, pattern fixed noise, and quantization error.

Sensor noise is composed of several different sources such as shot noise, dark current, fixed pattern noise, and quantization noise caused by the Analog-to-Digital

Converter. The process of generating an image starts with the photon flux on the sensor. Here, shot noise along the signal occurs and can distort the true intensity of received light. Shot noise is a result of the imperfections in the photodiode and create uncertainty in the number of photons captured in its respective pixel. The quantum efficiency (QE) of a sensor is a metric dictating approximately how much of the photon flux is converted to electrons. While the entire sensor may be qualified to a specific QE rating, each pixel may independently be more or less efficient, causing photo response non-uniformity, an additional noise source taking the form of a constant pattern of brightness or darkness perturbing the image. This is usually caused by the variance in the photodiode surface area. Dark current is a form of noise that is a result of residual electric current in the pixel wells while there is no incident light being projected on the focal plane and can be generated as long as the exposure is open and can worsen with age and temperature, among other factors. Dark current is also subject to shot noise and a fixed-pattern noise further distorting the image. At this stage, each pixel is charged according to its collected electrons and is converted to voltage. The conversion between electrons and voltage is often non-linear, causing further discrepancy between the incoming signal from the star and the registered values in the sensor. Some fixed pattern noise is also generated at this stage as a result of the imperfections in the amplifier for each column in the sensor. Finally, the voltage signal is transformed to a digital signal by means of an Analog-to-Digital Converter, or ADC. Because the ADC converts a continuous signal into a discrete signal, quantization error is introduced and affects the relative brightness values for the image.

To faithfully simulate the image received by a star tracker, it is important to consider how the noise is generated and by what means can it be modeled. Simulating noise in an optical sensor is non-trivial and requires several sensor-specific values given by its datasheet. Konnik et al. (2014) [24], proposes a system to sim-

ulate the corruption of an image via noise from several sources including shot noise, dark current, read noise, quantization error, etc. Following the model presented by Konnik, an image is simulated using the parameters described in Table 4.1 and the stars in the FOV of the star tracker for a given simulation; i.e., derived from the randomly generated right ascension and declination. Once an image has been simulated, a threshold of the mean image brightness plus 5 standard deviations, i.e., $\mu_{\text{image, brightness}} + 5\sigma_{\text{image, brightness}}$, as recommended by Liebe [11] is applied to filter out dimmer stars and prevent the measurement process for mistaking noise as a star. Due to being an inherently stochastic process, noise is generated on each simulation run as opposed to a blanket cap on the maximum star magnitude visible.

Image Generation

Before any noise is simulated, the ideal image is blurred by means of a gaussian filter with $\sigma = 2$ as recommended for star trackers by Liebe [11]. To build the image, suppose the image plane is composed of $n \times n$ discrete pixels and can be represented by a matrix, $I_{\text{image}} \in \mathbf{R}^{n \times n}$. A pixel reacts to photoelectron collisions and converts the energy into voltage which is read by the sensor and interpreted as brightness; more photoelectrons directly correlates to more brightness. To calculate the photoelectrons captured from a star with given magnitude, m , Liebe presents an expression:

$$e^- = Q_e \cdot \Phi_p \cdot \frac{1}{2.5^m} \cdot t_i \cdot A \quad (4.44)$$

where e^- represents the photoelectrons captured from a given star, $Q_e \cdot \Phi_p$ represents the total photon influx of a magnitude 0 star derived from the star photon flux and the sensor quantum efficiency, t_i represents the integration or exposure time, and A is the aperture area; in this case, Φ_p was determined to be $31,800e^-$ [11]. This can be applied to each star in the FOV for a given simulation to determine the

total photoelectrons received strictly from the starlight. Liebe also proposes that the image be intentionally blurred e.g., via a gaussian filter with σ of 2 to spread the starlight out over the pixels [11]. This alleviates the burden on the centroiding phase compared to if the starlight was concentrated on a single pixel. Without intentional blurring, concentrated light sources may fill and cause non-uniform blooming/spill in the neighboring pixels [32], skewing the centroid calculation. If, however, the starlight is purposefully and normally distributed over a pre-determined window size, e.g., a 7×7 -pixel region of interest, centroid accuracy was experimentally proven to increase with the square of a star's magnitude [11].

For this model, each star in the ideal image is represented as a 7×7 -pixel region defined by a gaussian distribution of the light received from a star as presented in Equation 4.44 where its position on the image plane is determined by the method presented by the $\Phi(\mathbf{S})$ model. Figure 4.2 presents the difference between a point-light and a defocused image; the latter representation will be used to model ideal stars in the image plane. Figure 4.3 presents a simulated image of stars without noise spanning several magnitudes without noise; the region of interest and σ were increased to 21×21 -pixels and 5, respectively, to easily identify stars for the reader.

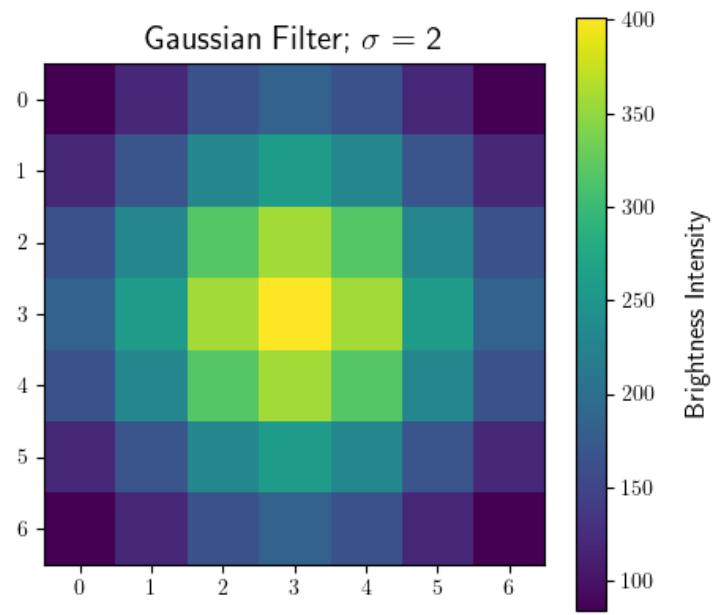
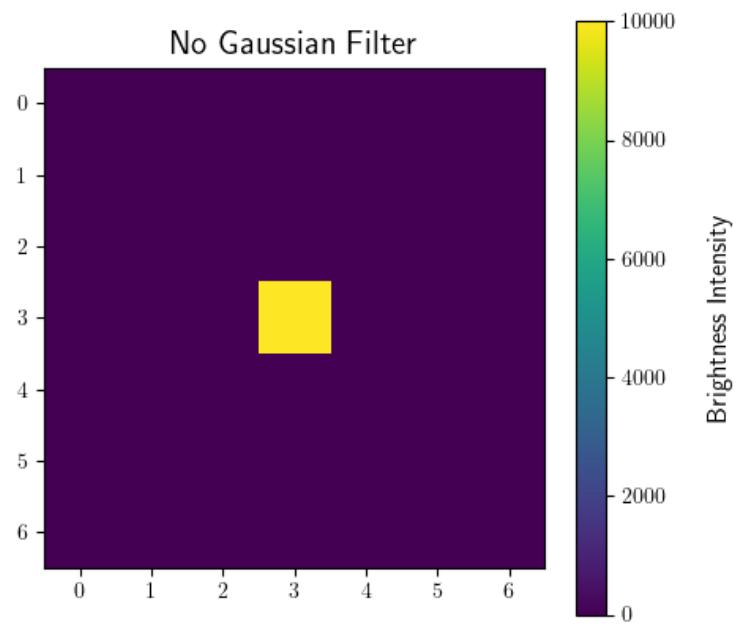


Figure 4.2: Unblurred star (above) and blurred star (below) by means of a Gaussian Filter with $\sigma = 2$, $ROI \in \mathbf{R}^{7 \times 7}$

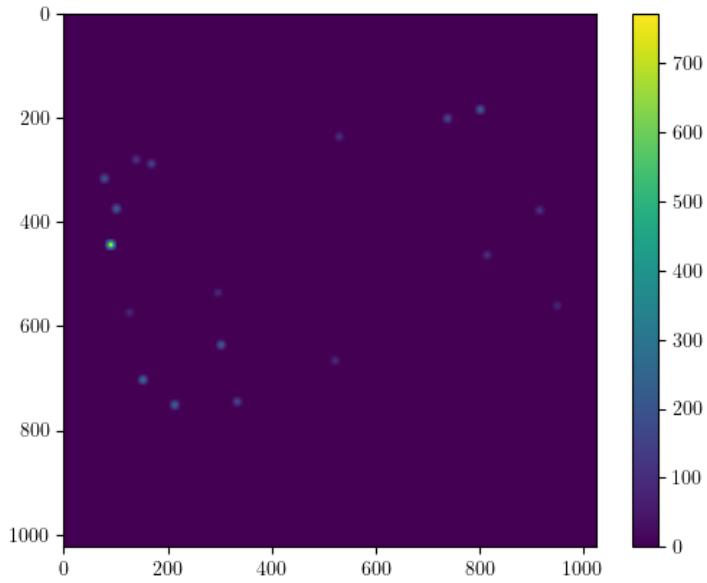


Figure 4.3: Sample simulated image with no noise; region of interest $\in \mathbf{R}^{21 \times 21}, \sigma = 5$ for visibility

To simulate noise, suppose a matrix, $I_{\text{noise},e^-} \in \mathbf{R}^{n \times n}$, representing the image plane where each element corresponds to the noise of the respective pixel. The noise model presented in Konnik et al. (2004) [24], is built by the physical process that takes place within the camera sensor, first by analyzing the captured photons from the stars, then by converting received power into registered electrons, then into a voltage read by the sensor, and finally into a digital signal mapping the brightness of the pixels.

Light Noise

The first source of noise is photon shot noise which is a result of the uncertainty of the number of photons collected from a given photodiode in the sensor. Shot noise is generated directly from the incoming starlight and can be modeled using the Poisson Distribution for low-light use-cases, $\mathcal{P}(\Lambda)$, where Λ represents the photon distribution

from the signal [24].

$$I_{\text{image} + \text{shot noise}} = \mathcal{P}(I_{\text{ideal}}) \quad (4.45)$$

While the photons received from the star can be entirely calculated based on the star's magnitude, the sensor is not 100% efficient in converting the energy into electrons. The quantum efficiency metric of a sensor for a given wavelength dictates how much of the incoming photons are registered and carried as electrons and can be directly applied to the image; i.e.,

$$I_{\text{image} + \text{QE}} = QE \cdot I_{\text{image}} \quad (4.46)$$

where QE is the scalar representation of the sensor's quantum efficiency, typically given at wavelength of ~ 550 nm.

Photo response non-uniformity is a source of noise stemming from the variations of the surface area of each photodiode - less surface area is subject to less photon collision and will register less brightness. Also called the PRNU, the noise source can be modeled as a zero-mean gaussian distribution with variance related to the PRNU factor, typically between 0.01 and 0.02 [24]. Notice, because the PRNU is signal-dependent, the additive noise is directly correlated to the respective pixel's brightness from the incoming starlight; i.e.,

$$I_{\text{image} + \text{PRNU}} = I_{\text{image}} + I_{\text{image}} \odot \mathcal{N}(0, \sigma_{\text{PRNU}}^2) \in \mathbf{R}^{n \times n} \quad (4.47)$$

where the $\square \odot \square$ operator denotes element-wise multiplication.

Once photon shot noise and PRNU are modeled, the next steps are to simulate dark noise, a thermally-induced noise source.

Dark Noise

Dark current is a source of noise independent from the incoming signal and arises from the irregularity of the sensor material and simulates electrons filling the pixel well. To simulate dark current, Konnik poses Equation 4.49 to determine the average dark current generated during a given exposure [24]. This can be modeled as a uniform noise generation from each pixel and raises the image's noise floor.

$$E_{\text{gap}}(T) = E_{\text{gap}}(0) - \frac{\alpha T^2}{T + \beta} \quad (4.48)$$

$$D_R = P_A D_{FM} T^{3/2} \exp\left(-\frac{E_{\text{gap}}}{2 \cdot kT}\right) \quad (4.49)$$

In Equation 4.48, E_{gap} , α , and β are each material constants with values of 1.1557[eV], $7.021 \cdot 10^{-4}$ [eV/K], and 1108[K], respectively. In Equation 4.49, P_A is the surface area of each pixel [cm^2] and D_{FM} is a sensor-specific performance metric for the dark current figure-of-merit [nA/cm^2] and relates the amount of current generated in the presence of heat at 300K [24]. The noise attributed to dark current can be modeled as a matrix of size $I_{\text{dark current}} = [D_R] \in \mathbf{R}^{n \times n}$.

Dark noise is also subject to shot noise and, similar to the photon shot noise, can be modeled with the Poission Distribution; i.e,

$$I_{\text{dark current} + \text{shot noise}} = \mathcal{P}(I_{\text{dark current}}) \quad (4.50)$$

Dark current Fixed Pattern Noise, or FPN, is noise stemming from the variability between pixels to generate noise. The FPN can also be modeled as a zero-mean gaussian distribution with variance of the FPN factor, typically between 10% and

40%, and is specific to each pixel; i.e.,

$$I_{\text{dark current} + \text{FPN}} = I_{\text{dark current}} + I_{\text{dark current}} \odot \mathcal{N}(0, \sigma_{\text{FPN}}^2) \in \mathbf{R}^{n \times n} \quad (4.51)$$

Via simulation, it was determined that the mean and standard deviation of dark noise in photoelectrons was 151.648 and 47.250 (1σ), respectively; this value was used to distort the image of ideal stars to minimize computation per simulation state. The full signal registered by the sensor is comprised of the incoming light from the star, noise from the incoming photons, and the thermally-generated noise from dark current; i.e.,

$$I_{\text{signal}} = I_{\text{starlight} + \text{noise}} + I_{\text{dark current}} \quad (4.52)$$

and will be treated as a single matrix of size $\mathbf{R}^{n \times n}$ to convert into voltage and eventually into brightness values. It is also imperative to set the bounds for I_{signal} such that the minimum value is 0, the maximum value is given by the saturation capacity of the sensor, e_{\max}^- , and each pixel has an integer number of electrons; i.e.,

$$I_{\text{signal}} = \min(I_{\text{signal}}, 0)$$

$$I_{\text{signal}} = \max(I_{\text{signal}}, e_{\max}^-)$$

$$I_{\text{signal}} = \lfloor I_{\text{signal}} \rfloor$$

Figure 4.4 superimposes the discussed light and dark noise on Figure 4.3 to illustrate the magnitude of noise on the image.

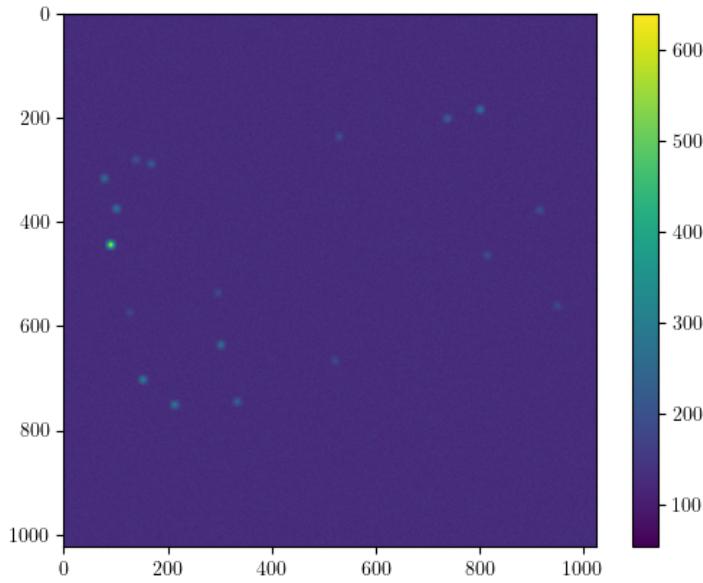


Figure 4.4: Sample simulated image with light and dark noise superimposed; region of interest $\in \mathbf{R}^{21 \times 21}$, $\sigma = 5$ for visibility

Charge to Voltage

Converting the charge from electrons into voltage for the sensor to read is non-trivial and requires several sensor-specific parameters such as the bus voltage, reset factor, and sense node gain. Prior to reading the electron charge, each pixel resets its values to determine a reference voltage for which to compare the charge; noise, however, is introduced at the reset step and is a function of the temperature and sense capacitance; i.e.,

$$\sigma_{\text{reset}} = \sqrt{\frac{k_B T}{C_{SN}}} \quad (4.53)$$

where C_{SN} is given by the charge of a single electron (1.6022e-19 [C]) divided by the sense node gain; i.e., q/A_{SN} . The reset noise, however, cannot be modeled as a normally distributed value; instead, the distribution can be modeled by taking the

exponential of the normal distribution; i.e.,

$$V_{\text{reset noise}} = \exp(\mathcal{N}(0, \sigma_{\text{reset}}^2)) - 1 \quad (4.54)$$

The reset noise can then be added to the signal voltage while taking into account the reset factor, a multiplier describing the amount of noise, on average, the sensor generates during reset. It should be noted that the increase in voltage corresponds to a decrease in brightness; i.e., more electrons decreases the respective pixel's voltage level. To convert charge into voltage, the model can be transcribed as

$$V_{\text{ref}} + (V_{\text{reset noise}} \cdot R_f) - (I_{\text{signal}} \cdot A_{SN}) \quad (4.55)$$

where R_f represents the reset factor.

Offset fixed pattern noise can be added to the signal and is a result of the columns in an image sharing an amplifier to read the signal. Each amplifier generates some noise, independent of the other amplifiers, whose intensity is directly correlated to the respective pixel's brightness. The offset fixed pattern noise can be modeled as being normally distributed with zero mean and a variance of σ_U , the amount of voltage generated by the amplifiers for a full-well pixel, typically 0.5% [24]; i.e.,

$$\sigma_U = 5 \cdot 10^{-4} \cdot V_{\text{FW}} \quad (4.56)$$

$$V_{\text{offset FPN}} = \mathcal{N}(0, \sigma_U^2) \in \mathbf{R}^{1 \times n} \quad (4.57)$$

$$V_{\text{signal voltage + FPN}} = V_{\text{signal voltage}} + \begin{bmatrix} V_{\text{offset FPN}} \\ \vdots \\ V_{\text{offset FPN}} \end{bmatrix} \in \mathbf{R}^{n \times n} \quad (4.58)$$

Notice, $I_{\text{offset FPN}}$ is a row vector of randomly distributed values that is repeated n times to simulate the column-constant noise added to the signal.

Voltage to Digital Numbers

At this stage, the full signal represents the incident star light and noise represented as a voltage; to convert the signal into digital values, an ADC, or, analog-digital converter, is required. An ADC converts the continuous voltage domain into discrete digital values where the sensitivity, or minimum voltage recognized per discrete value, is given by the number of bits in the converter. If there are N bits in an ADC, 2^N unique values can be represented and the resolution is given by the difference in the voltage domain divided by the number of discrete values; i.e.,

$$K_{\text{ADC}} = \frac{V_{FW} - V_{\min}}{2^N} \quad (4.59)$$

where V_{\min} represents the minimum quantifiable voltage and V_{FW} represents the voltage reading for a full pixel well. These values can also be represented by the following expressions:

$$V_{FW} = \frac{e_{\max}^- \cdot q}{C_{SN}} \quad (4.60)$$

$$V_{\min} = \frac{q \cdot A_{SN}}{C_{SN}} = A_{SN}^2 \quad (4.61)$$

where q represents the elementary charge of an electron.

While Equation 4.59 gives the resolution for the ADC; i.e., the voltage per discrete step, a more useful metric to convert voltage into digital signal is its inverse; i.e., the digital signal per voltage. This metric is simply given by the inverse of Equation 4.59; i.e.,

$$A_{\text{ADC}} = \frac{2^N}{V_{FW} - V_{\min}} \quad (4.62)$$

Finally, the ADC gain given in Equation 4.62 can be applied to the signal voltage to determine the digital signal for each pixel; i.e,

$$DN_{\text{signal}} = A_{\text{ADC}} \cdot V_{\text{signal}} \quad (4.63)$$

$$DN_{\text{signal}} = \text{round}(DN_{\text{signal}}) \quad (4.64)$$

Rounding, as in Equation 4.64, is a necessary step as a digital signal only supports integer values; quantization error occurs here as values between integers are artificially reduced or increased depending on their proximity to the nearest integer. Figure 4.5 superimposes the noise during the conversion from electrons to a digital signal on Figure 4.4.

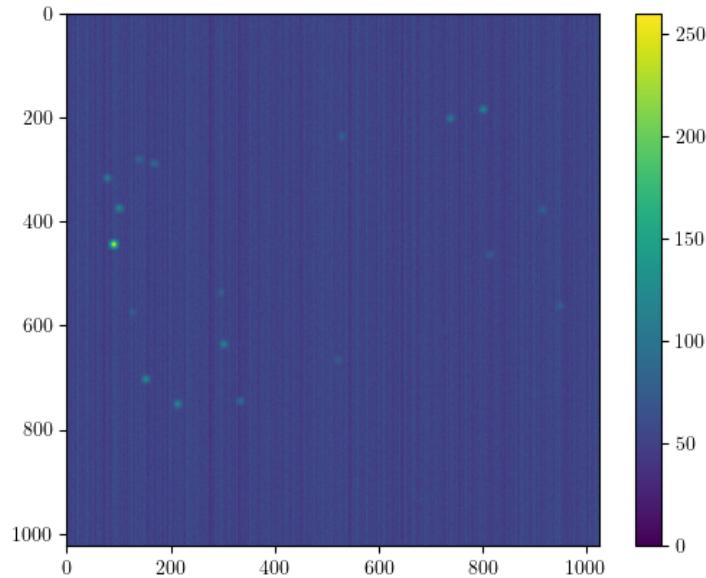


Figure 4.5: Sample simulated image with conversion noise superimposed; region of interest $\in \mathbf{R}^{21 \times 21}, \sigma = 5$ for visibility

The result of simulating the noise is an image with the incident starlight corrupted by sensor deficiencies. An artificial threshold, recommended by Liebe, can be applied

in software by setting any pixel less than a predetermined value, typically the mean brightness plus 5 standard deviations, to 0 to reduce noise. This alleviates some computation by the processor and ensures that the noise artifacts are not mistakenly identified as a star. Equation 4.65 presents the process of applying the threshold to the output signal of the image. Figure 4.6 applies the threshold to Figure 4.5 for comparison and Figure 4.7 presents another sample image before and after the threshold is applied.

$$P_{ij} \in DN_{\text{signal}} \begin{cases} P_{ij} & \text{if } P_{ij} \geq \mu_{\text{signal}} + 5\sigma_{\text{signal}} \\ \text{else} & 0 \end{cases} \quad (4.65)$$

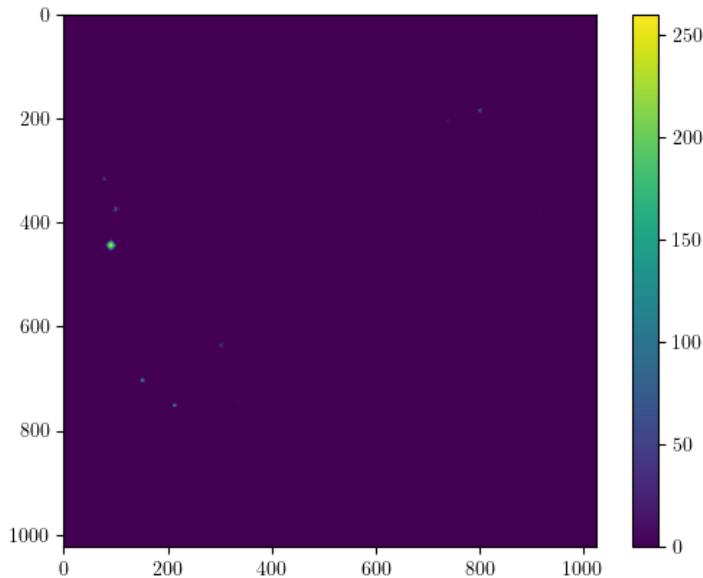


Figure 4.6: Sample simulated image with threshold applied; region of interest $\in \mathbf{R}^{21 \times 21}$, $\sigma = 5$ for visibility

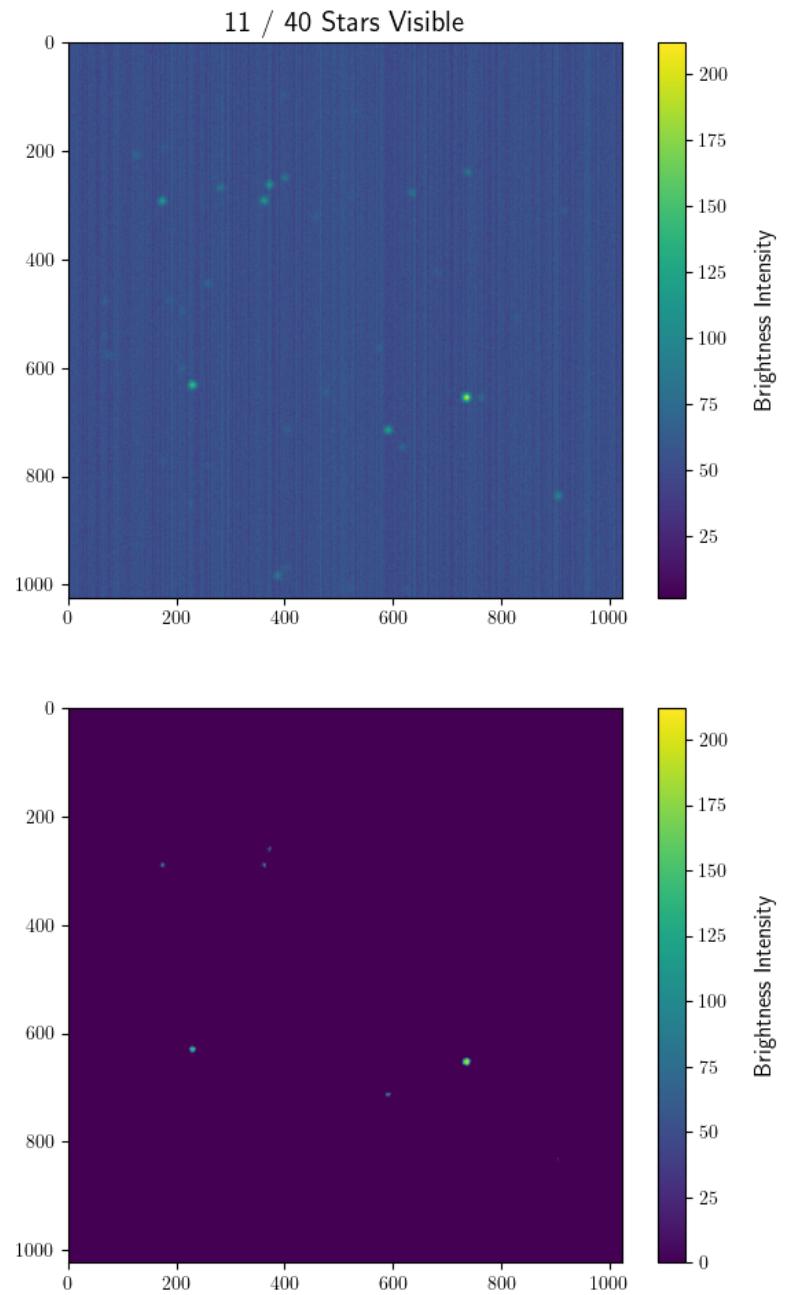


Figure 4.7: Before (above) and after (below) applying the brightness threshold; $\text{ROI} \in \mathbb{R}^{21 \times 21}$, $\sigma = 5$ and signal boosted by factor of 5 for visibility

4.2.3 Centroiding Accuracy

The process of centroiding plays a critical role in star tracker measurements, where the goal is to accurately determine the positions of stars within images corrupted by noise. By employing advanced algorithms, centroiding analyzes these images and identifies the precise locations of each star. Additionally, it enables the creation of unit vectors associated with the stars, thereby transforming the image into a valuable dataset of celestial information. To model centroiding error, information from the previous steps during the hardware aberration and noise simulation stage can be utilized to faithfully recreate how centroids and the measured star vectors are computed without the need of a centroiding implementation.

Suppose the measured star vector, $\hat{\mathbf{S}}$, was computed and produces an image on the plane at (x_π^*, y_π^*) as described in Section 4.2.1. The measured star centroid, $(\tilde{x}_\pi, \tilde{y}_\pi)$, can be artificially set with some normal distribution centered at the true centroid; i.e.,

$$\begin{bmatrix} \tilde{x}_\pi \\ \tilde{y}_\pi \end{bmatrix} = \begin{bmatrix} x_\pi^* \\ y_\pi^* \end{bmatrix} + \begin{bmatrix} \eta_x \\ \eta_y \end{bmatrix} \quad (4.66)$$

where (η_x, η_y) represent the difference between the measured and true star centroid. The measured star vector, $\tilde{\mathbf{S}}$, can then be recomputed as in Equation 3.2.

$$\tilde{\mathbf{S}} = \frac{1}{\sqrt{\tilde{x}_\pi^2 + \tilde{y}_\pi^2 + F^2}} \begin{bmatrix} -\tilde{x}_\pi \\ -\tilde{y}_\pi \\ F^2 \end{bmatrix} \quad (4.67)$$

Liebe suggests that the centroiding accuracy is directly related to the intensity of the star; i.e., centroids of brighter stars tend to be more accurately computed

as compared to dimmer stars. Figure 4.8, adapted from Liebe [11], illustrates the relation between centroiding accuracy and the intensity of the signal measured in electrons for the $\sigma = 2$ case [11].

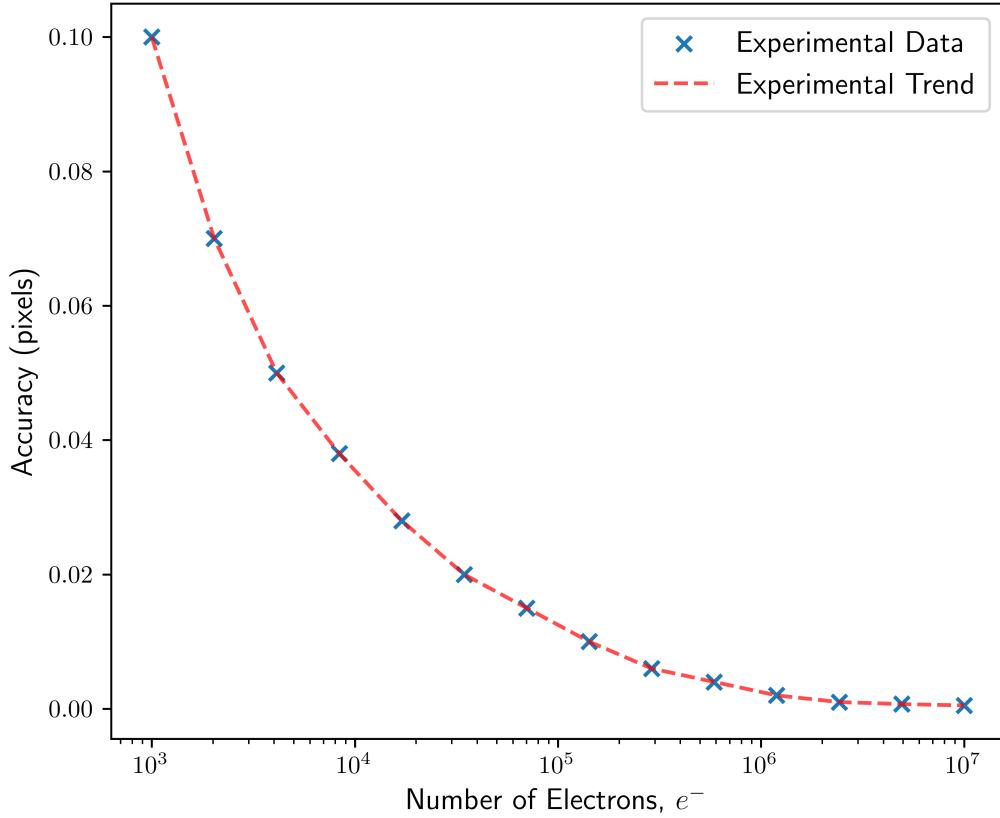


Figure 4.8: Centroid accuracy versus number of electrons with Gaussian filtering $\sigma=2$

To reevaluate the results in terms of the apparent magnitude of the star, the input must be transformed from the electron count to the magnitude. Equation 4.44 presents the relation between a star's magnitude, m and the collected electrons, e^- , and is reprinted below.

$$e^-(m) = Q_e \cdot \Phi_p \cdot \frac{1}{2.5^m} \cdot t_i \cdot A$$

The inverse of Equation 4.44 can be constructed such that the magnitude can be analytically determined given the total electron count; the inverse is presented in Equation 4.68.

$$m(e^-) = \log_{2.5} \left(\frac{Q_e \cdot \Phi_p \cdot t_i \cdot A}{e^-} \right) \quad (4.68)$$

Using the data provided by Liebe, the relation between the star and centroiding accuracy can be reinterpreted as the centroiding accuracy as a function of the magnitude. Figure 4.9 presents the new data directly transcribed and transformed from Figure 4.8.

Liebe proposes that the model between star magnitude and centroiding accuracy is exponential in nature [11]; to compute the estimated model, a regression minimizing the R^2 , or variance, can be computed. After applying the regression with the data presented in Figures 4.8 and 4.9, it was determined that the model for centroiding accuracy as a function of magnitude, $C(m)$, can be expressed by the following equation:

$$C(m) = 0.00333124 \cdot e^{0.406843m} - 0.00153084 \quad (4.69)$$

with an associated R^2 value of 0.. The $C(m)$ model can be used to independently determine the values for η_x and η_y for each star in each simulation. The centroiding model, $C(m)$, describes the typical centroiding error and the exact value can be modeled by a normal distribution, i.e., $\eta = \mathcal{N}(0, C(m)^2)$. It can be assumed that centroiding error is equally likely to happen in any direction relative to the true centroid. A random parameter, θ , can be used to determine the direction of error which is recomputed for each star in a single simulation. Equation 4.70 presents how θ can be generated from a uniform probability density to dictate the values for η_x

and η_y for each star.

$$\theta = U(0, 2\pi) \quad (4.70)$$

$$\eta = \mathcal{N}(0, C(m)^2) \quad (4.71)$$

$$\eta_x = \eta \cos \theta \quad (4.72)$$

$$\eta_y = \eta \sin \theta \quad (4.73)$$

For reference, the $C(m)$ model implies that for a star of magnitude 7, the associated centroiding accuracy is approximately 0.190 pixels and for a star of magnitude 0, the accuracy is approximately 0.021 pixels. Figure 4.9 presents the centroid model regression and the associated experimental data from Liebe.

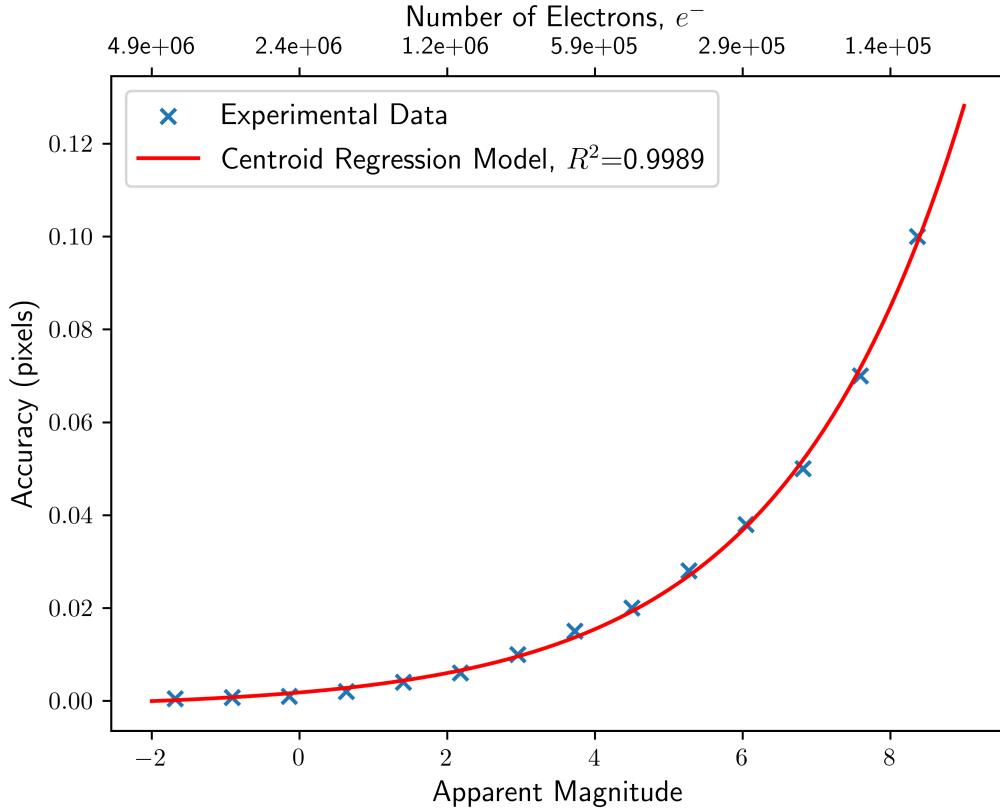


Figure 4.9: Centroid accuracy versus apparent magnitude, $R^2 = 0.9989$

4.3 Effects of the Space Environment

In addition to the major sources of error presented by Read Eisenman et al. (1998) [27], and Jia et al. (2010) [26], the space environment may worsen conditions for the star tracker. In LEO, the main cases to study the effects of the space environment can be broken down into the thermal environment and the radiation environment. These factors can pose an engineering challenge for CubeSats and can worsen the accuracy for the onboard star tracker.

4.3.1 Thermal Environment

For star trackers in particular, the lens and sensor quality is of interest when analyzing thermal cycles. The model results expressed in Chapter 5.3 used an average temperature, T_0 , of 25C° as the baseline for computation. In LEO, however, space-craft can experience a large temperature range which can cause thermal stresses on all hardware.

Thermal changes can cause lens deformation, leading to an unobservable change in the focal length and additional distortion of the camera vectors. Jamieson et al. (1981) thoroughly describes the relation between optical systems and thermal effects, especially noting how the focal length changes with temperature [33] and arrives at an expression denoting the effective opto-thermal expansion coefficient, x_f :

$$x_f = \frac{1}{f} \frac{df}{dT} = x_g - \frac{1}{n - n_{air}} \left(\frac{dn}{dt} - n \frac{dn_{air}}{dt} \right) \quad (4.74)$$

where f is the focal length, x_g is the change of surface radii of the lens with respect to temperature, n is the refractive index at a given temperature, n_{air} is the refractive index of air $\equiv 1.0$. Notice, the expansion coefficient, x_f , is a function of the focal length and the ambient temperature and has no units. Jamieson compiles a table of

opto-thermal expansion coefficients for popular lens materials where it was determined that the distribution follows a normal curve and the mean and standard deviation was 7e-6 and 3.9e-5, respectively; i.e.,

$$x_f \sim \mathcal{N}(7e-6, 3.9e-5^2)$$

The distribution given by Equation 4.75 can be used to randomly assign a simulated star tracker an opto-thermal expansion coefficient to simulate the effects of temperature swings.

In addition to its effect on the focal length and distortion, the thermal environment also affects the noise generated by the sensor during the conversion between photons to a digital signal. As given by Equations 4.48-4.49 and 4.53, the temperature has a direct effect on the dark current and reset noise. Depending on whether more or less noise is generated, the result of the environment could raise the threshold to remove dimmer stars or lower the threshold and cause the system to mistake an especially bright spot as a star. Fortunately, the Pyramid Algorithm is highly robust to false star identifications and was experimentally determined to not be an issue [20].

To simulate the environment's effect on change in focal length and noise generation, the temperature should be faithfully modeled for LEO CubeSats. Garzón et al. (2018) presents a high fidelity model to simulate the temperature of a 3U CubeSat in LEO. The temperature was derived from first principles governed by the law of thermodynamics and considers sources of the heat flux from the Earth albedo, Earth infrared radiation, solar radiation, and reflected solar radiation. It was presented that a 3U CubeSat with typical material properties can experience a sinusoidal temperature range between 0C° and 60C° depending on the time of the year and the satellite's position with respect to the Sun [34].

To simulate the effects on accuracy, the opto-thermal expansion coefficient given by Equation 4.75 is added to the distribution of ϵ_z as presented in Equation 4.76 and can be considered as another source of image plane translation error which can be accurately modeled by the $\Phi(\mathbf{S})$ model.

$$\epsilon_z = \mathcal{N}(0, \sigma_{\epsilon_z}^2) + (T - T_0) \cdot F \cdot x_f \quad (4.75)$$

$$\epsilon_z = \mathcal{N}(0, 0.5^2) + (T - 25) \cdot F \cdot \mathcal{N}(7e - 6, 3.9e - 5^2) \quad (4.76)$$

The noise generation can also be updated with the new temperature range to generate more realistic noise scenes. The results of the change in focal length and noise generation are given below.

4.3.2 Radiation Environment

To model the changes in dark current, a multiplier can be attached to the expression presented in Equation 4.49 and is presented in Equation 4.77. This multiplier simulates how radiation increases dark current across the image. A similar approach to model the increase in fixed pattern noise can also be applied, resulting in Equation 4.78 derived from Equation 4.51.

$$D_R = 150\% \cdot P_A D_{FM} T^{3/2} \exp\left(-\frac{E_{gap}}{2 \cdot kT}\right) \quad (4.77)$$

$$I_{\text{dark current} + \text{FPN}} = I_{\text{dark current}} + (200\% \cdot I_{\text{dark current}} \odot \mathcal{N}(0, \sigma_{\text{FPN}}^2)) \quad (4.78)$$

Responsivity pertains to a pixel's ability to convert photoelectron energy into a digital signal. Hopkinson conducted an experiment in which it was observed that the average responsivity of sensors decreased by around 30% following exposure to irradiation levels comparable to those encountered in LEO [23]. This decrease can be mathematically represented by attenuating the received signal of the sensor by

30%. In other words, the expression described in Equation 4.52 can be reformulated as Equation 4.79 to simulate the reduced responsiveness of the sensor in question.

$$I_{\text{signal}} = 0.7 \cdot (I_{\text{starlight}} + \text{noise} + I_{\text{dark current}}) \quad (4.79)$$

Other effects caused by radiation, such as threshold voltage shift and ADC performance, were experimentally determined to not be major concerns [35].

4.4 QUEST

Once the measured star vectors are generated, the model simplifies the measurement process by asserting that if a star is registered, it is also identified according to the catalog. This simplification implies that there are no failure-to-identifies or false identifications to distort the final attitude estimation. This is a valid assumption and is corroborated by the experimental data collected by Mortari et al. (2004), for the Pyramid Algorithm [20] which is highly robust to false identifications and highly accurate for images with at least 4 stars.

The final step in the measurement process is to determine the rotation between the body-fixed vectors and the identified inertial vectors; i.e., using the information to solve Wahba's Problem. The QUaternion ESTimator algorithm, or QUEST algorithm, is a popular choice for its simplicity and accuracy on flight computers [11]. QUEST accepts a set of vectors in one frame and the corresponding vectors in another and computes the optimal rotation via computing the optimal eigenvector associated with the optimal eigenvalue of Wahba's Problem [12]. Algorithm 4 presents one implementation of the QUEST algorithm [12].

Applying the QUEST algorithm is the final step in the measurement process model. Recall, each simulation state contains the true rotation via the right ascension,

Algorithm 4 QUaternion ESTimator Algorithm

Require: $|\vec{\mathbf{V}}_{\oplus}| > 1$
Require: $|\vec{\mathbf{V}}_{\oplus}| \equiv |\vec{\mathbf{V}}_{\star}|$

```

function F_LAMBDA( $\lambda, a, b, c, d, k_{22}$ )
     $f(\lambda) = \lambda^4 - (a + b)\lambda^2 - c\lambda + (ab + ck_{22} - d)$ 
    return  $f(\lambda)$ 
end function

function FPRIME_LAMBDA( $\lambda, a, b, c$ )
     $f'(\lambda) = 4\lambda^3 - 2(a + b)\lambda - c$ 
    return  $f'(\lambda)$ 
end function

function OPTIMALLAMBDAMULTIPLIER( $a, b, c, d, k_{22}$ )
     $\lambda_0 = 1$ 
     $\lambda_1 = \lambda_0 - F\_Lambda(\lambda_0, \dots) / FPrime\_Lambda(\lambda_0, \dots)$ 
    while  $|\lambda_1 - \lambda_0| \geq TOL$  do ▷ Newton's Iteration Solver
         $\lambda_0 = \lambda_1$ 
         $\lambda_1 = \lambda_0 - F\_Lambda(\lambda_0, \dots) / FPrime\_Lambda(\lambda_0, \dots)$ 
    end while
    return  $\lambda_1$ 
end function

function QUEST( $\vec{\mathbf{V}}_{\oplus}, \vec{\mathbf{V}}_{\star}$ )
     $\mathbf{B} = \sum \left( \vec{\mathbf{V}}_{\oplus} \cdot \vec{\mathbf{V}}_{\star}^T \right)^T$ 
     $\mathbf{k}_{12} = [B_{23} - B_{32}, B_{31} - B_{13}, B_{12} - B_{21}]$ 
     $k_{22} = \text{tr}[\mathbf{B}]$ 
     $\mathbf{S} = \mathbf{B} + \mathbf{B}^T$ 
     $a = k_{22}^2 - \text{tr}[\text{adj}[\mathbf{S}]]$ 
     $b = k_{22}^2 + \mathbf{k}_{12}^T \cdot \mathbf{k}_{12}$ 
     $c = |\mathbf{S}| + \mathbf{k}_{12}^T \cdot \mathbf{S} \cdot \mathbf{k}_{12}$ 
     $d = \mathbf{k}_{12}^T \cdot \mathbf{S}^2 \cdot \mathbf{k}_{12}$ 
     $\lambda = \text{OptimalLambdaMultiplier}(a, b, c, d, k_{22})$ 
     $\alpha = \lambda^2 - k_{22}^2 + \text{tr}[\text{adj}[\mathbf{S}]]$ 
     $\beta = \lambda - k_{22}$ 
     $\gamma = (\lambda + k_{22}) \cdot \alpha - |\mathbf{S}|$ 
     $\mathbf{X} = (\alpha \cdot \mathbf{I}^{3 \times 3} + \beta \cdot \mathbf{S} + \mathbf{S}^2) \cdot \mathbf{k}_{12}$ 
     $f = \sqrt{\gamma^2 + \mathbf{X}^T \cdot \mathbf{X}}$ 
     $\epsilon = \mathbf{X} / f$ 
     $\eta = \gamma / f$ 
     $\mathbf{q}_{\text{est}} = [\epsilon \ \eta]^T$ 
    return  $\mathbf{q}_{\text{est}}$ 
end function

```

declination, and roll angle. Equations 4.80 - 4.85 can be applied to the true angles to determine the true quaternion which can be compared to the estimated rotation given by QUEST. Equation 2.29 gives the accuracy of the output quaternion.

$$\mathbf{C}_{\star\oplus} = \mathbf{R}_z(\phi)\mathbf{R}_y(\delta - \pi/2)\mathbf{R}_z(-\alpha) \quad (4.80)$$

$$\eta = \frac{(\text{tr}[\mathbf{C}] + 1)^{\frac{1}{2}}}{2} \quad (4.81)$$

$$\epsilon_1 = \frac{C_{23} - C_{32}}{4\eta} \quad (4.82)$$

$$\epsilon_2 = \frac{C_{31} - C_{13}}{4\eta} \quad (4.83)$$

$$\epsilon_3 = \frac{C_{12} - C_{21}}{4\eta} \quad (4.84)$$

$$\mathbf{q}_{\text{true}} = [\epsilon_1 \ \epsilon_2 \ \epsilon_3 \ \eta]^T \quad (4.85)$$

The QUEST Algorithm will determine the accuracy of the star tracker, and by extension, the effects of the input parameters to distort the star tracker model.

4.5 Gamma Function

The Gamma Function, as referenced in Algorithm 2, is the accumulation of the error models characterizing the star tracker measurement process and describes how the input parameters are transformed to estimate the accuracy of the star tracker. The Gamma Function applies the distortions in the order of which they appear during the measurement process to accurately capture the effects of compounding errors and begins with the image capturing stage. The $\Phi(\mathbf{S})$ model is used to distort the incoming star vectors before then applying the noise model to determine which stars are bright enough for the star tracker to register. Once the stars are selected based on their apparent magnitude, its position is determined on the image plane and artificially skewed as to simulate the centroiding accuracy. Once the measured star positions

and measured star vectors are computed, it is assumed each of their corresponding inertial representations can be determined and is subsequently passed into the QUEST Algorithm to determine the measured rotation. The measured quaternion is compared to the true quaternion derived from the input parameters and an accuracy value is returned. This process repeats several times to faithfully capture the distributions of each input and their effects on the final accuracy estimate of the star tracker. Algorithm 5 outlines the specific process to transform the input parameters into an estimated accuracy¹.

The purpose of the measurement process model is to be able to independently modulate various parameters; i.e., according to some distribution, and gauge the effects on the accuracy of the star tracker. The Gamma Function returns the estimated accuracy of the star tracker given the potential state in the form of the magnitude of error between the estimated quaternion and the true quaternion. It is expected for the star tracker measurement to be normally distributed about the true measurement and where its accuracy can be presented by the standard deviation of its estimates. The model, however, returns the magnitude of the error which is strictly positive. Consequentially, the estimated accuracy derived from the star tracker model follows a half-normal distribution, a special case of the normal distribution where the data is strictly positive (or negative); i.e., $Y \sim |\mathcal{N}(\mu_{FN}, \sigma_{FN}^2)|$ [36], where Y is the set of data representing the star tracker accuracy estimate and \mathcal{N} represents the underlying normal distribution function. The standard deviation of the distribution relates to the precision of the star tracker; that is, with lower standard deviation, precision increases and vice versa. For true half-normal distributions, i.e., for distributions centered around 0, the standard deviation can be computed as presented in Equation 4.86. However, for distributions centered around a non-zero value, the normal

¹A star tracker requires at least two stars be visible to acquire an attitude estimate. With a single star, there does not exist a unique representation of the spacecraft attitude nor will the identification method be capable of identifying the star [11].

distribution is more appropriate to use for modeling where the mean of the data represents the expected bias in the measurement. The standard deviation for normally distributed data can be calculated according to Equation 4.88 where μ represents the mean of the data. Their probability distribution functions are presented in Equations 4.87 and 4.89, respectively [36].

$$\sigma = \sqrt{\frac{\sum x_i^2}{n}} \quad (4.86)$$

$$f(x; \sigma) = \frac{\sqrt{2}}{\sigma\sqrt{\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (4.87)$$

$$\sigma = \sqrt{\frac{\sum(\mu - x_i^2)}{n}} \quad (4.88)$$

$$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{x - \mu}{\sigma}\right)^2\right) \quad (4.89)$$

The goal of the measurement process model presented in this paper is to form concepts and relations between the specific input parameters and the expected accuracy of the star tracker which can be measured by the standard deviation of the set of estimated accuracy values. The model requires some knowledge of the star tracker system which should faithfully represent what is typical for low-cost star trackers. A set of star tracker properties have been identified and presented in Table 4.1 to evaluate the effects of star tracker errors. The presented values is considered standard for COTS star trackers [11] [24] [30], although the model is built independent of the parameters.

Most star tracker operate between 10 and 20 degrees of FOV and use a sensor typically between 256 and 1024 pixels across [11]. For a standard 1024×1024 image sensor and $3.5\mu\text{m}$ pixel size, a 12mm focal length lens was selected, resulting in a focal length of approximately 3500 pixels and a 16.6 degree FOV. The sensor parameters

were determined to be typical for modern CMOS sensors and have been chosen to reflect the current standards for noise generation and simulation [24]. The centroid model, as presented in Section 4.2.3, was derived from the experimental data presented by Liebe on the relation between centroiding accuracy and apparent star magnitude. A simulation determined that with the sensor parameters described in Table 4.1, the mean maximum magnitude discernable was 5.39 with a standard deviation of 0.115 (1σ). To simplify the simulation model, an additional parameter is introduced modulating the maximum magnitude visible to a star tracker in a given image.

Each simulation represents a single feasible state the star tracker can exhibit and returns the error in its estimated attitude compared to the true attitude. The distribution of errors is presented and represents the estimated star tracker accuracy.

Model Star Tracker			
	Property	μ	1σ
Optics	Focal Length, px	3500	0
	Sensor Size, px	1024	0
	ϵ_x , px	0	0.1
	ϵ_y , px	0	0.1
	ϵ_z , px	0	0.1
	ϕ_x , $^\circ$	0	0.01
	θ_y , $^\circ$	0	0.01
	ψ_z , $^\circ$	0	0.01
Sensor	Pixel Length, μm	3.5	0
	Quantum Efficiency, %	60	0
	0 Magnitude Photon Flux, e^-	32,000	0
	Integration Time, s	0.2	0
	Region of Interest, px^2	49	0
	Gaussian Smoothing, σ	2	0
	PRNU Factor	0.01	0
	Dark Noise Factor	0.1	0
	Pixel Saturation Level, e^-	10,000	0
Software	Maximum Magnitude Visible	5.39	0.115
Software	Centroid Accuracy, px	0	$C(m)$

Table 4.1: Model star tracker properties used for analysis

While the model star tracker characterized by Table 4.1 will serve as the basis for the examination of error propagation, each component will also be evaluated independently to provide insight on the effects at the component level on attitude estimation.

Algorithm 5 Measurement Process Model

X : Feasible star tracker state from PDF, $[g_1(\xi) \dots g_n(\xi)]^T$
 α : Right Ascension
 δ : Declination
 ϕ : Roll Angle

```

function  $\Gamma(X, \alpha, \delta, \phi)$ 
   $\mathbf{q}_{\text{true}} = \text{AnglesToQuaternion}(\alpha, \delta, \phi)$   $\triangleright$  True rotation representation
   $\Lambda = \text{StarProjection}(\alpha, \delta, \phi)$   $\triangleright$  List of visible star and ECI Vectors
  if Number of Stars in  $\Lambda \leq 1$  then
    return FAIL  $\triangleright$  See Footnote 1
  end if

   $\mathbf{C}_{\star\oplus} = \mathbf{R}_z(\phi)\mathbf{R}_y(\delta - \pi/2)\mathbf{R}(-\alpha)$   $\triangleright$  Create ECI to Body-Fixed Rot. Mat.
   $\text{Image} \leftarrow \mathbf{0} \in \mathbf{R}^{w \times h}$   $\triangleright$  Initialize empty image matrix
  for each  $\vec{\mathbf{V}}_{\oplus}$  in  $\Lambda$  do
     $\vec{\mathbf{V}}_{\star} = \mathbf{C}_{\star\oplus} \cdot \vec{\mathbf{V}}_{\oplus}$ 
     $\vec{\mathbf{V}}_{\text{meas}} = \Phi(\vec{\mathbf{V}}_{\star})$   $\triangleright$  Compute measured star vectors
     $x_{\pi}^*, y_{\pi}^* \leftarrow \text{Image Position from } \Phi(\vec{\mathbf{V}}_{\text{meas}})$ 
     $\sigma_x^2, \sigma_y^2 = C(m)/\sqrt{2}$   $\triangleright$  Compute centroiding accuracy for each star
     $\tilde{x}_{\pi} = x_{\pi}^* + \mathcal{N}(0, \sigma_x^2)$ 
     $\tilde{y}_{\pi} = y_{\pi}^* + \mathcal{N}(0, \sigma_y^2)$ 
     $\text{Star} \leftarrow \text{generateStarSignal}(\tilde{x}_{\pi}, \tilde{y}_{\pi})$ 
     $\text{Image} \leftarrow \text{Image} + \text{Star}$   $\triangleright$  Add star signal to image
  end for

   $\text{Image} \leftarrow \text{NoiseSimulation}(X)$   $\triangleright$  Simulate noise in the image
   $B_{\min} = \mu_{\text{Image}} + 5\sigma_{\text{Image}}$   $\triangleright$  Set brightness threshold
   $\{\vec{\mathbf{V}}_{\oplus}\} \leftarrow \emptyset$ 
   $\{\vec{\mathbf{V}}_{\star}\} \leftarrow \emptyset$   $\triangleright$  Initialize empty sets to store ECI and Body-Fixed Vectors
  for each  $\vec{\mathbf{V}}_{\oplus}$  in  $\Lambda$  do
    if  $B(\vec{\mathbf{V}}_{\oplus}) \geq B_{\min}$  then
       $\{\vec{\mathbf{V}}_{\oplus}\} \leftarrow \vec{\mathbf{V}}_{\oplus}$ 
       $\{\vec{\mathbf{V}}_{\star}\} \leftarrow [-\tilde{x}_{\pi}, -\tilde{y}_{\pi}, F]^T$   $\triangleright$  Store vectors if above threshold
    end if
  end for

   $\mathbf{q}_{\text{est}} = \text{QUEST}(\{\vec{\mathbf{V}}_{\oplus}\}, \{\vec{\mathbf{V}}_{\text{vis}}\})$ 
   $\vartheta = |\mathbf{q}_{\text{true}} \otimes \mathbf{q}_{\text{est}}^*|$ 
  return  $\vartheta$ 
end function
  
```

Chapter 5

SENSITIVITY AND MULTIVARIATE ANALYSIS

5.1 Analysis Approach

The goal of the measurement process model presented in this paper is to identify the effects of key parameters on star tracker accuracy. To begin, a comprehensive Monte Carlo analysis of the star tracker parameterized by Table 4.1 is presented and examined. Subsequently, the constituent parameters of the star tracker are extracted and analyzed for their individual effects on accuracy. This progression enables an examination of the translation, rotation, and centroiding errors individually and by doing so, the contribution of each parameter to the overall system's performance is uncovered. Lastly, the space environment and various fringe cases are examined, providing a comprehensive understanding of the star tracker's performance in its entirety.

Ideal Case

To validate the model, a Monte Carlo simulation was derived whereby each input parameter was held to its ideal value; i.e., no errors are propagated through the system. The model should return an exactly 0 arcsecond accurate star tracker representing the ideal case at each step in the measurement process. Based on the simulation results presented in Figure 5.1, it can be seen that this supposition holds for the most part. The minimal error presented is likely a result of floating point limitations and predetermined simulation tolerances stopping before an optimal solution is reached, such as the `OptimalLambdaMultiplier` function in the QUEST Algorithm which employs the Newton's Iteration Method in determining a suitable lambda multiplier.

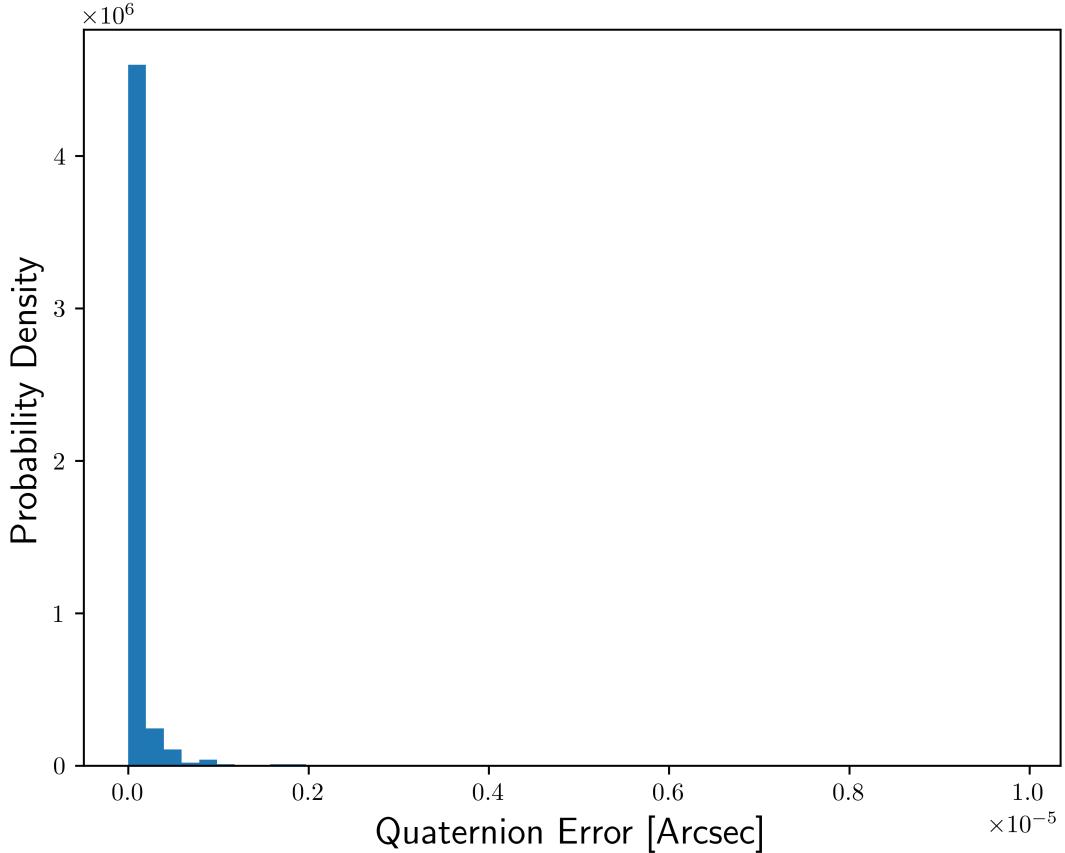


Figure 5.1: Attitude accuracy for a star tracker with ideal parameters

5.2 Multivariate Analysis

By performing the simulation as presented in Algorithms 3 and 2 and modulating each parameter discussed in Table 4.1, the distribution for attitude estimate error can be determined. Figure 5.2 presents this distribution and, contrary to the initial assumption that the data should be half-normal, it is clear that there is some unexpected skew in the data. The mean of the distribution, 11.601 arcseconds, represents the typical offset or bias the star tracker exhibits in its attitude estimates and, conversely, the standard deviation, 28.376 arcseconds, represents the accuracy of the star tracker. This suggests that the star tracker measurement will typically fall within approximately 28 arcseconds of an off-nominal target by 12 arcseconds.

In other words, if the true attitude is represented as \mathbf{q}_{true} , then the star tracker will return a measurement that is accurate within 28 arcseconds, 1σ , to \mathbf{q}_{est} which in and of itself is 12 arcseconds away from \mathbf{q}_{true} .

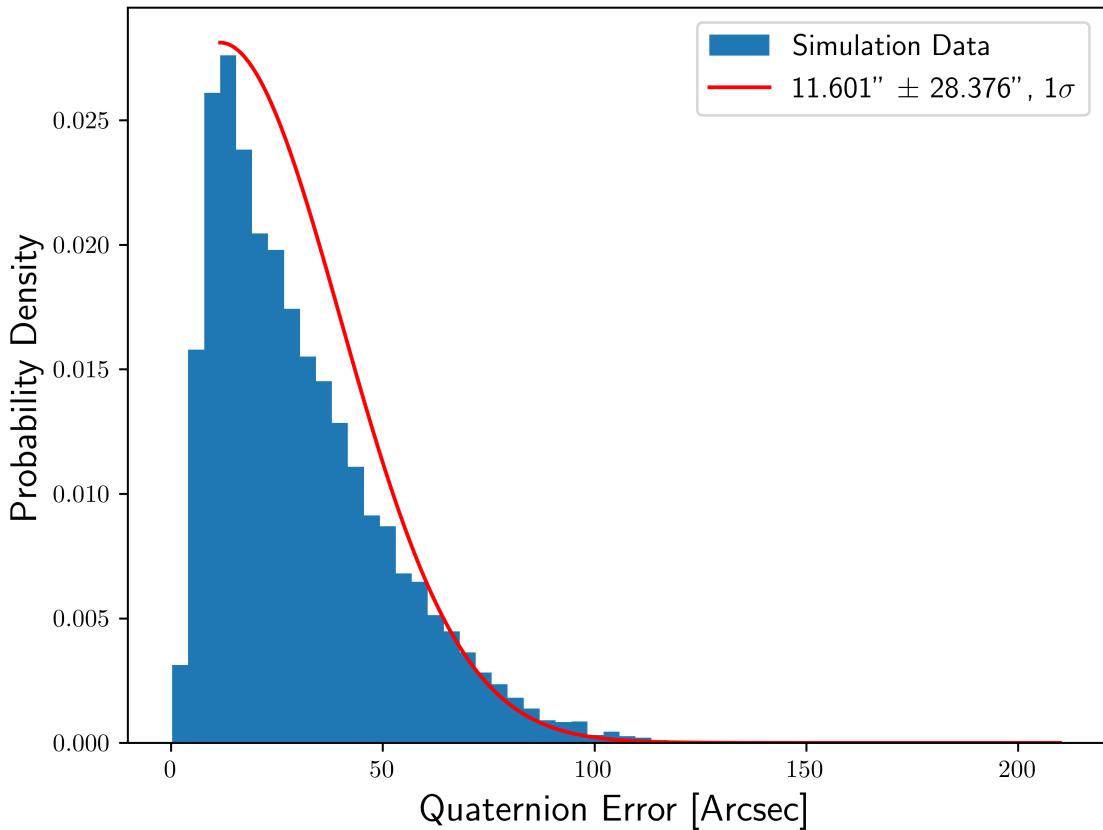


Figure 5.2: Attitude accuracy (11.601 ± 28.376 arcseconds, 1σ) with all inputs varied

It is important to understand that the star tracker analyzed for Figure 5.2 is comprised of the errors in the image plane translation, rotation, and centroiding errors. To understand the sources of the bias and accuracy, it is crucial to decompose the model into its constituent parameters and analyze its effects.

Hardware Aberration

Beginning with the effects from the hardware, Figures 5.3 and 5.4 present the distribution of error for a star tracker subject to solely 3 degrees of image plane translation and 3 degrees of image plane rotation, respectively. Based on the presented distributions, it can be seen that the source for the bias in the star tracker originates from the translation in the image plane whereas the large accuracy range originates from the rotation in the image plane. Something to note is that the accuracy of the star tracker is not necessarily the sum of the accuracy of its constituents. In part, this can be attributed to conflicting errors negating the effects of each other.

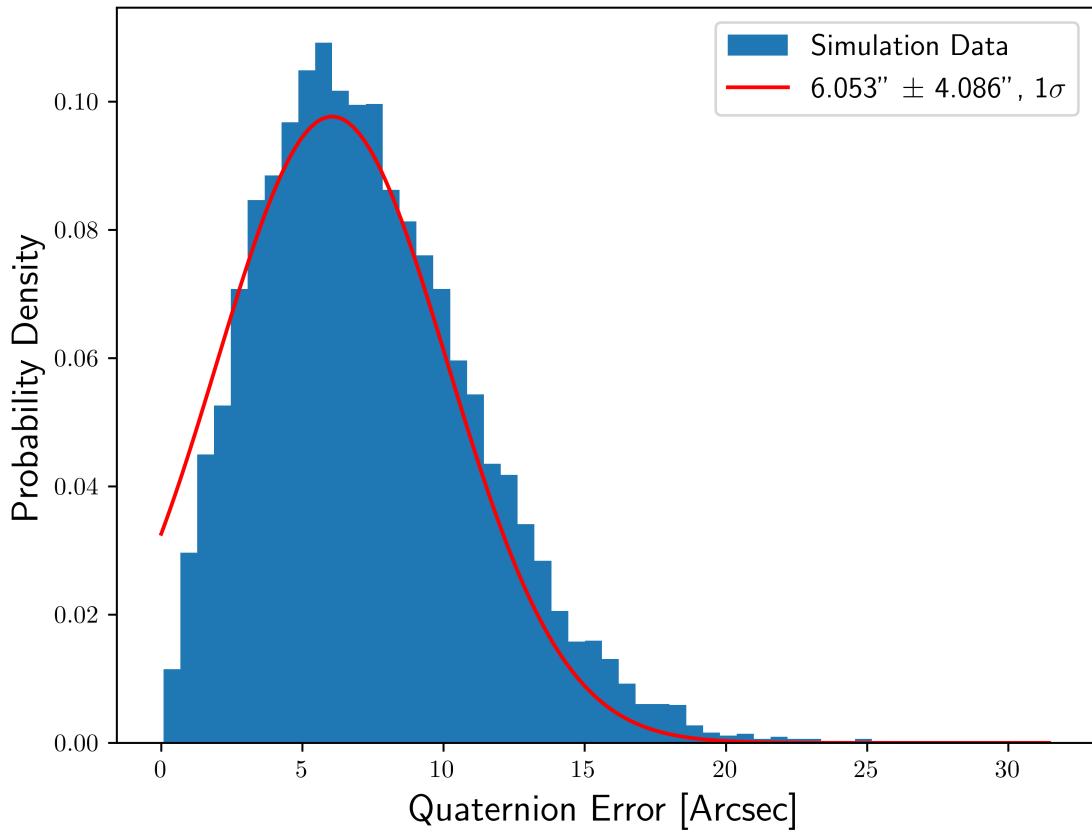


Figure 5.3: Attitude accuracy (6.053 ± 4.086 arcseconds, 1σ) versus 3 degrees of image plane translation

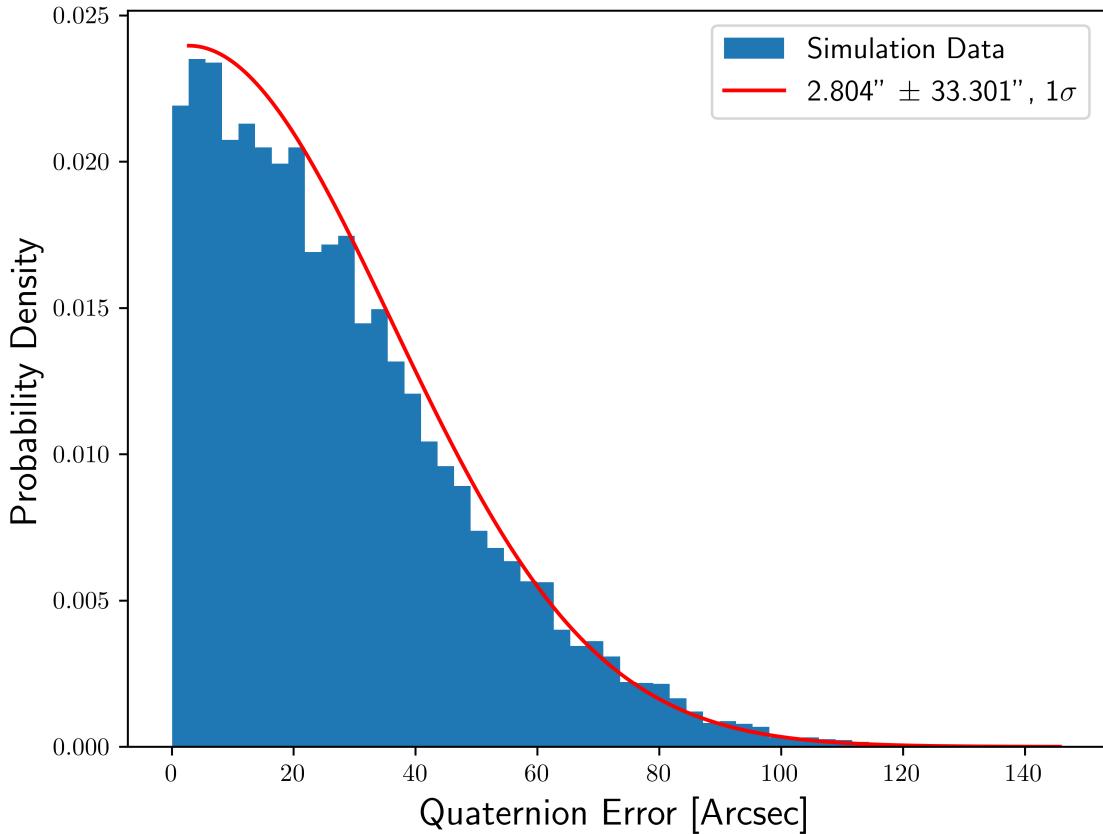


Figure 5.4: Attitude accuracy (2.804 ± 33.301 arcseconds, 1σ) versus 3 degree image plane rotation

These results are expected; by translating the image plane away from the center, the boresight is effectively shifted and the computation for the attitude is done with respect to a phantom-boresight, or its ideal location. This shift is consistent between measurements which causes the consistent bias in the attitude estimates. Conversely, by inducing a rotation in the image plane, the star vectors are directly distorted which can cause large errors in the attitude estimates. These effects can further be analyzed by further decomposing these parameters into their lateral and vertical counterparts.

Figure 5.5 presents the distribution or error by analyzing solely the translation of the image plane in the lateral axes and Figure 5.6 presents the effects by examining the translation of the image plane in the vertical direction. From here, it is made

clear that the effects of translation in the lateral axes is what provides the bias in the star tracker measurement. This supports the conjecture previously made where the attitude estimates are formed against a phantom-boresight as opposed to its true position. By injecting a bias of approximately 6.3 arcseconds with an accuracy of 4 arcseconds, the lateral translation is a concern and should be addressed to minimize star tracker error.

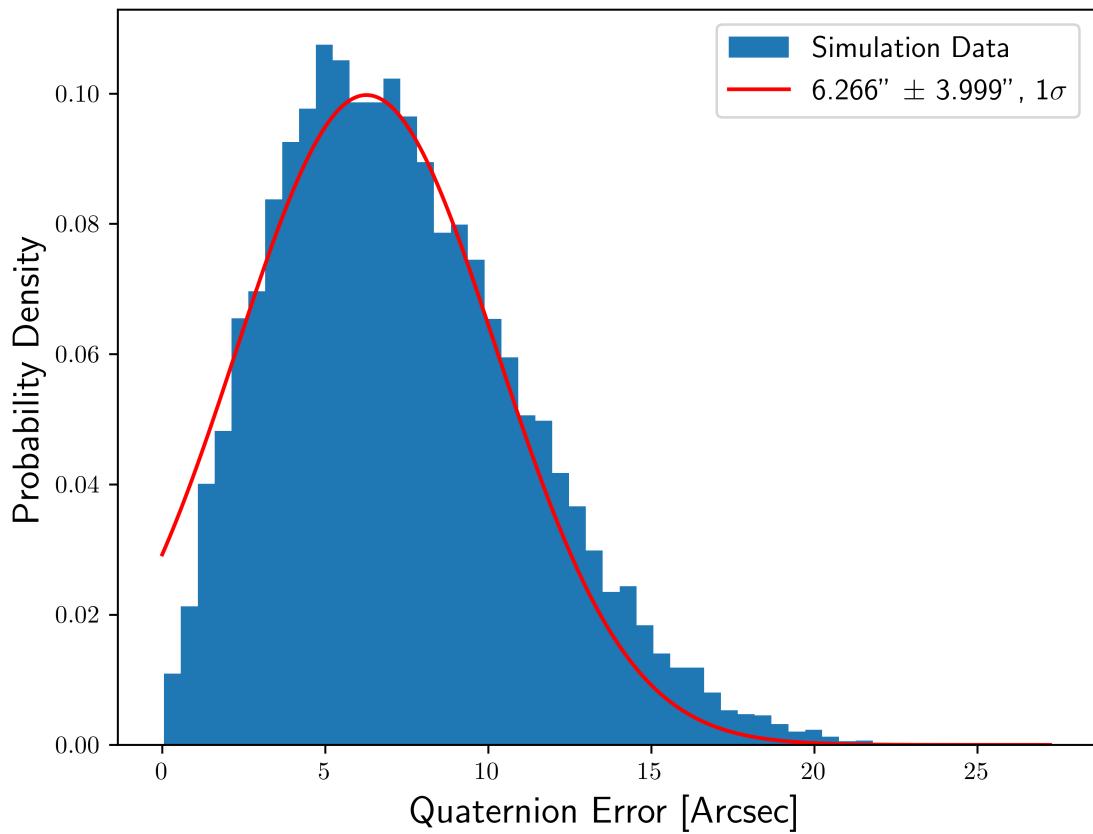


Figure 5.5: Attitude accuracy (6.266 ± 3.999 arcseconds, 1σ) versus lateral image plane translation

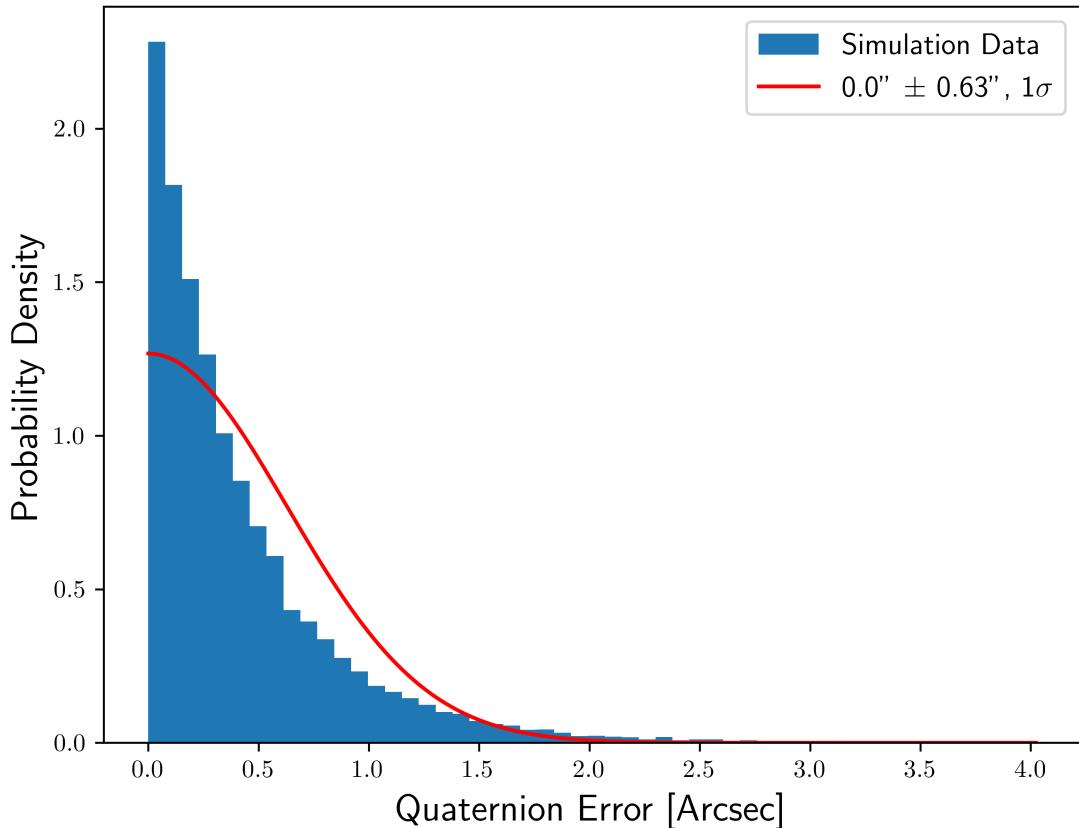


Figure 5.6: Attitude accuracy (0.000 ± 0.630 arcseconds, 1σ) versus vertical image plane translation

The sensitivity of the star tracker with respect to lateral translation can also be determined via a sensitivity analysis presented in Figure 5.7. By following the contours in the figure, the sensitivity analysis suggests that for every 0.1 pixels of translation, there is an approximately 5 arcsecond reduction in accuracy. Figure 5.8 presents a similar analysis for translation in the vertical direction. By comparing Figures 5.7 and 5.8, it is clear that the effects of lateral translation far outweigh the effects of vertical translation. While the contours for the vertical translation are much more shallow, indicating an approximate 0.05 arcsecond error for each 0.1 pixels of unaccounted translation, the measurements become increasingly uncertain, as evidenced by the probability density expanding as the translation grows. The lack of large bias is likely due to the absence of shifting the boresight in the vertical

translation case. The shift in focal length caused by the vertical translation would also modify the effective FOV of the star tracker. While this may pose a problem for capturing stars in a single image, the impact on the distortion of the star vectors is minimized and therefore the error caused by this is also relatively low compared to its lateral counterpart.

Fortunately, image plane translation is likely due to a manufacturing defect and, consequentially, is likely to remain constant throughout the lifetime of the star tracker. As a result, this hardware aberration can likely be calibrated prior to launch to minimize the bias in the attitude measurements.

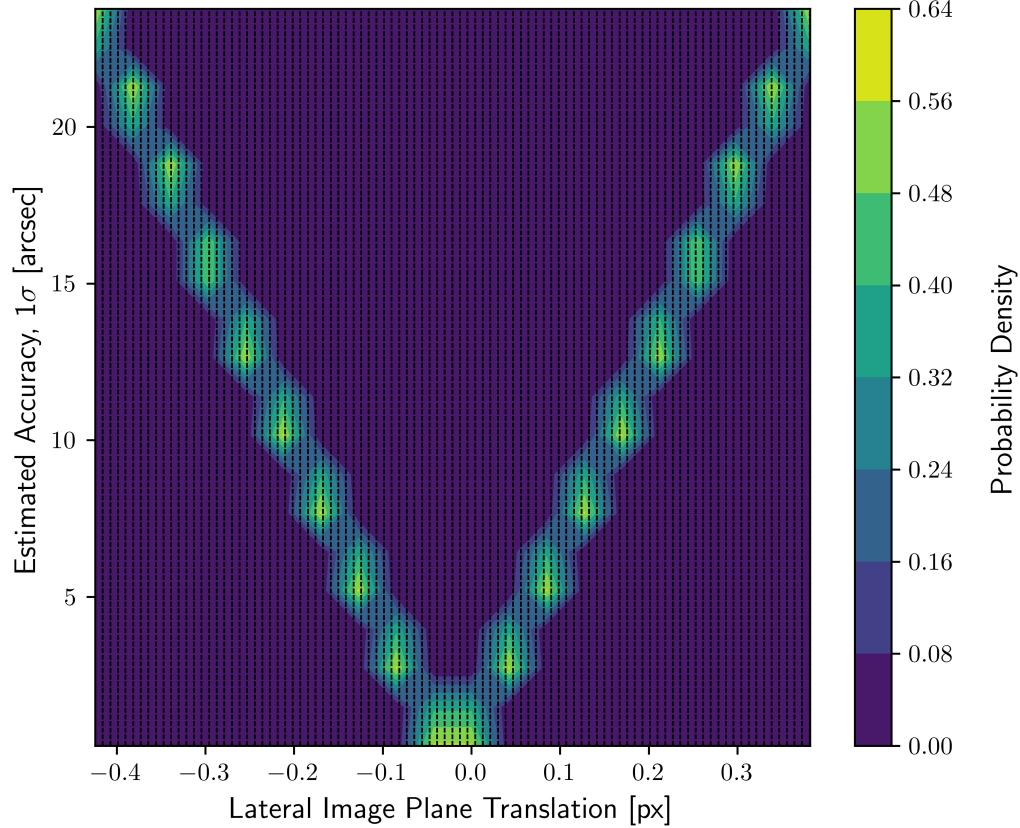


Figure 5.7: Sensitivity of accuracy with respect to lateral image plane translation

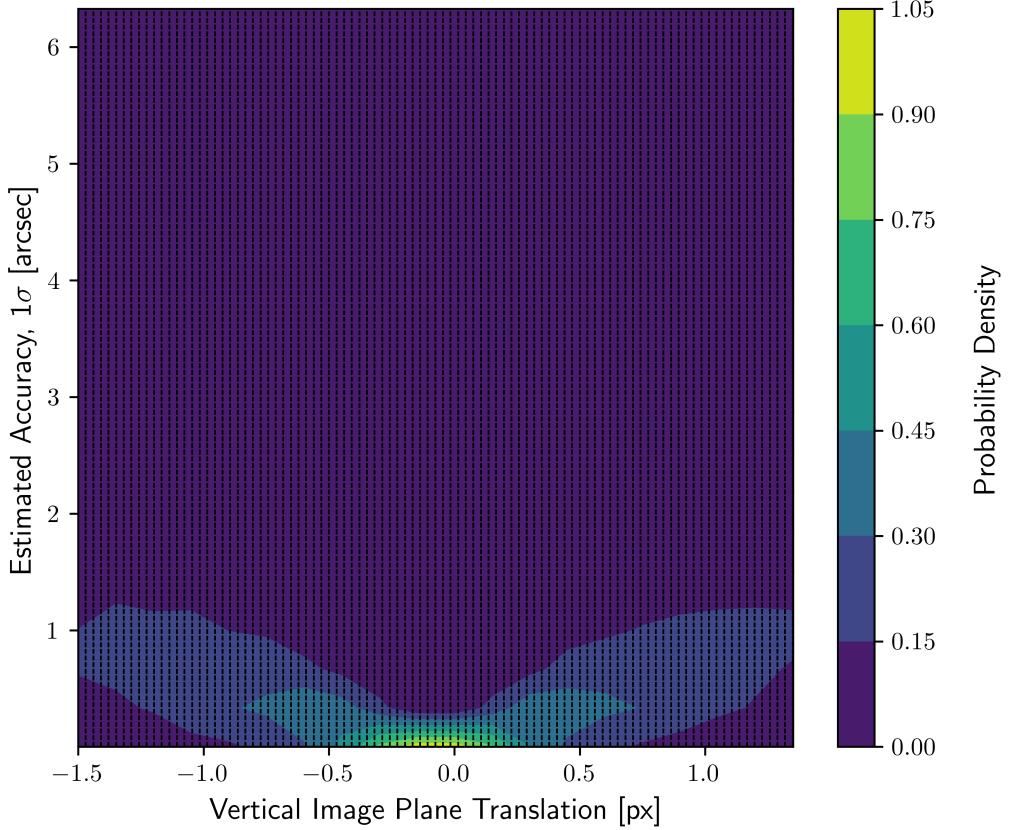


Figure 5.8: Sensitivity of accuracy with respect to vertical image plane translation

Separate from the image plane translation, the star tracker is also subject to rotation in the image plane. Figure 5.9 presents the distribution of error by examining the effects of rotation of the image plane about its lateral axes. Unlike its translational counterpart, by only inducing approximately 0.097 ± 0.329 (1σ) arcseconds of error, rotation about the lateral axes does not induce large errors in the attitude estimates. This is likely due to the preservation of the star vectors relative to each other. However, unlike rotation in the lateral axes, inducing a rotation about the vertical axis, or boresight, induces significant error. Figure 5.10 presents the distribution of error by analyzing the effects of rotation about the vertical axis, indicating an error of 0.002 ± 35.877 (1σ) arcseconds. Notably, the distribution reveals that the error is proportional to the rotation angle. This should make sense as rotating

the image plane about its boresight is equivalent to inducing additional artificial roll in the star tracker. The distribution upholds the validity of the model by describing how a rotation angle sampled from a distribution $\mathcal{N}(0, 0.01^{\circ}2)$ results in an error of 36 arcseconds (1σ); recall that 1 degree is equivalent to 3600 arcseconds.

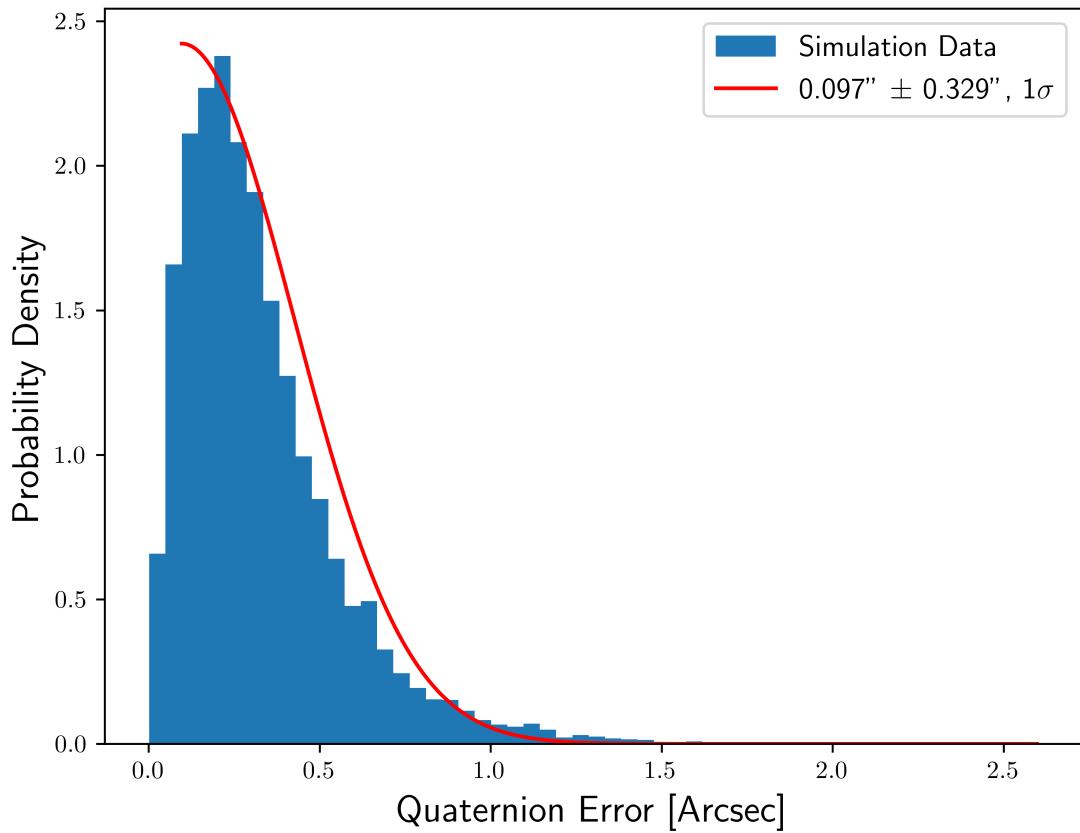


Figure 5.9: Attitude accuracy (0.097 ± 0.329 arcseconds, 1σ) versus image plane rotation about lateral axes

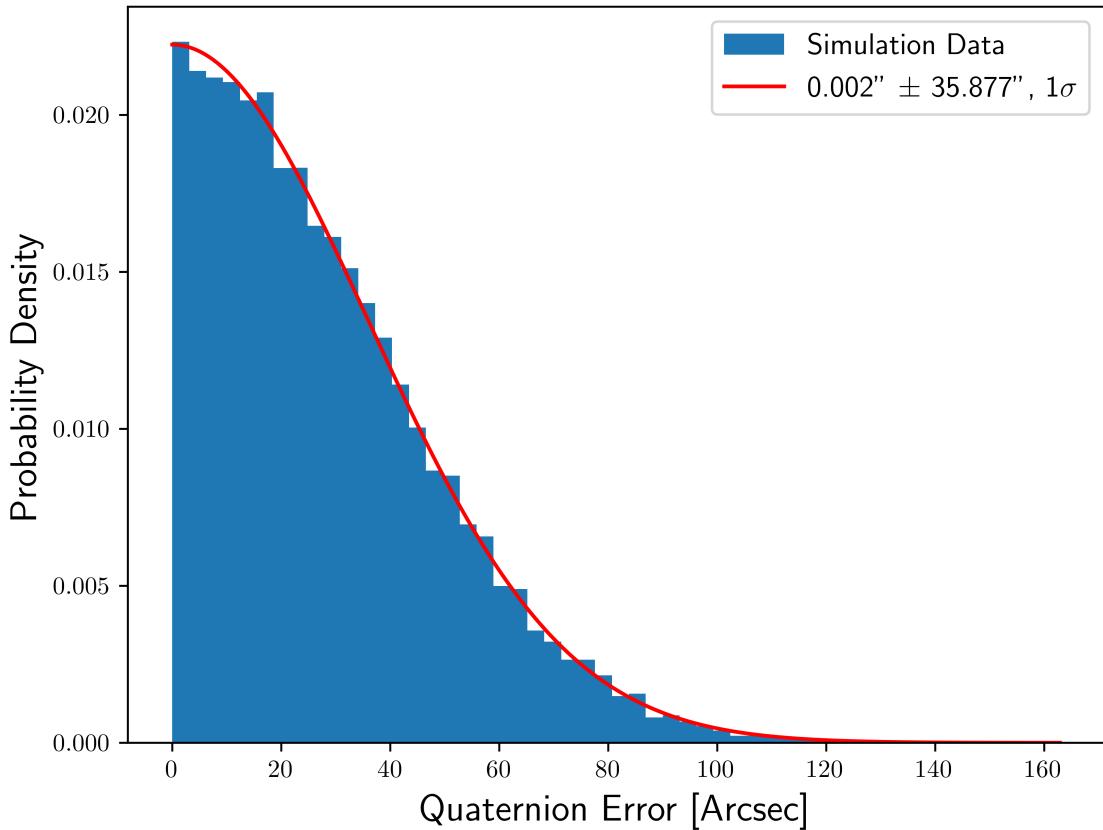


Figure 5.10: Attitude accuracy (0.002 ± 35.877 arcseconds, 1σ) versus image plane rotation about vertical axis

The sensitivity of each rotation type can also be quantified by examining the contours presented in Figures 5.11 and 5.12 which present the approximate error induced by rotating the image plane about its lateral axes and its vertical axis, respectively. Interestingly, the trends found in the effects of image plane translation are inverted for the effects of image plane rotation; that is, rotation about the lateral axes are shown to have much less of an effect on star tracker error whereas error is highly sensitive to rotation about the vertical axis. Figure 5.11 suggests that for every 0.01 degrees of rotation about the lateral axes, error is increased by approximately 0.185 arcseconds. Similar to the trend found for vertical translation, as the magnitude of image plane rotation increases, as does the uncertainty of the measurement. When solving Wahba's Problem, it is likely that the error residual is directly proportional to

the magnitude of rotation in the lateral axes, resulting in a wide variety of rotations that satisfy the cost function. Conversely, the Figure 5.12 indicates that there is a clear relation between error and rotation about the vertical axis and that the effect is highly sensitive. As discussed earlier, for every 0.01 degrees of rotation about the vertical axis, there is an additional error of 36 arcseconds; this scenario is analogous to registering additional roll in the system when there is none.

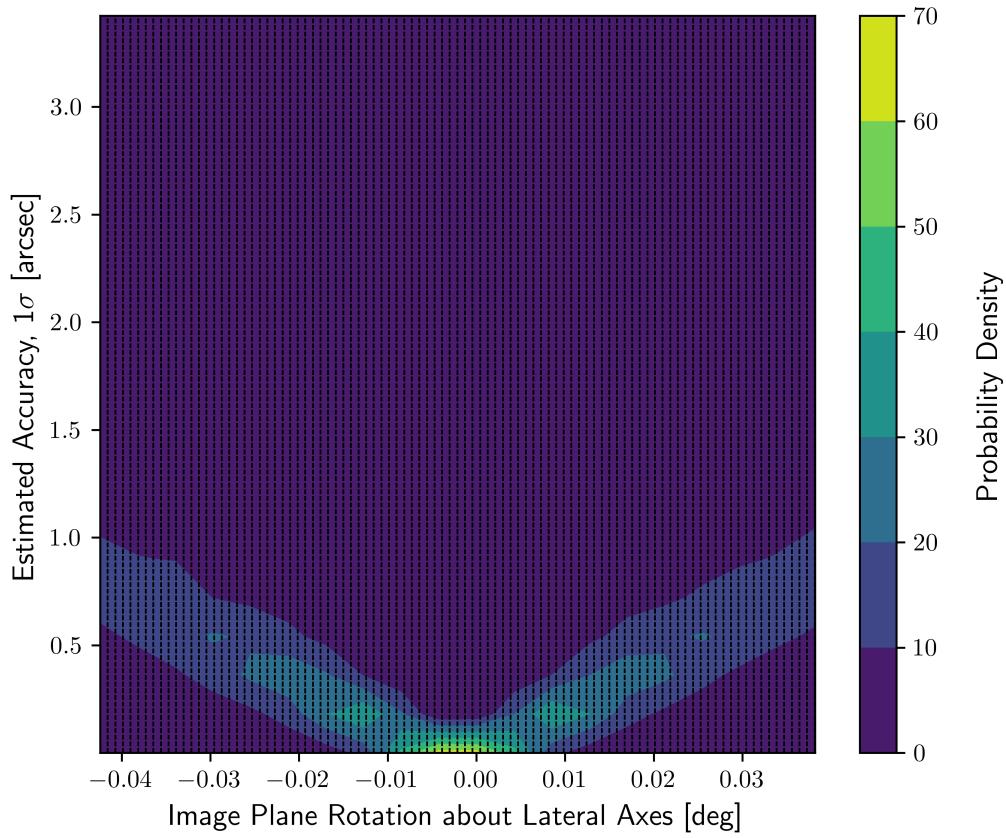


Figure 5.11: Sensitivity of accuracy with respect to image plane rotation about lateral axes

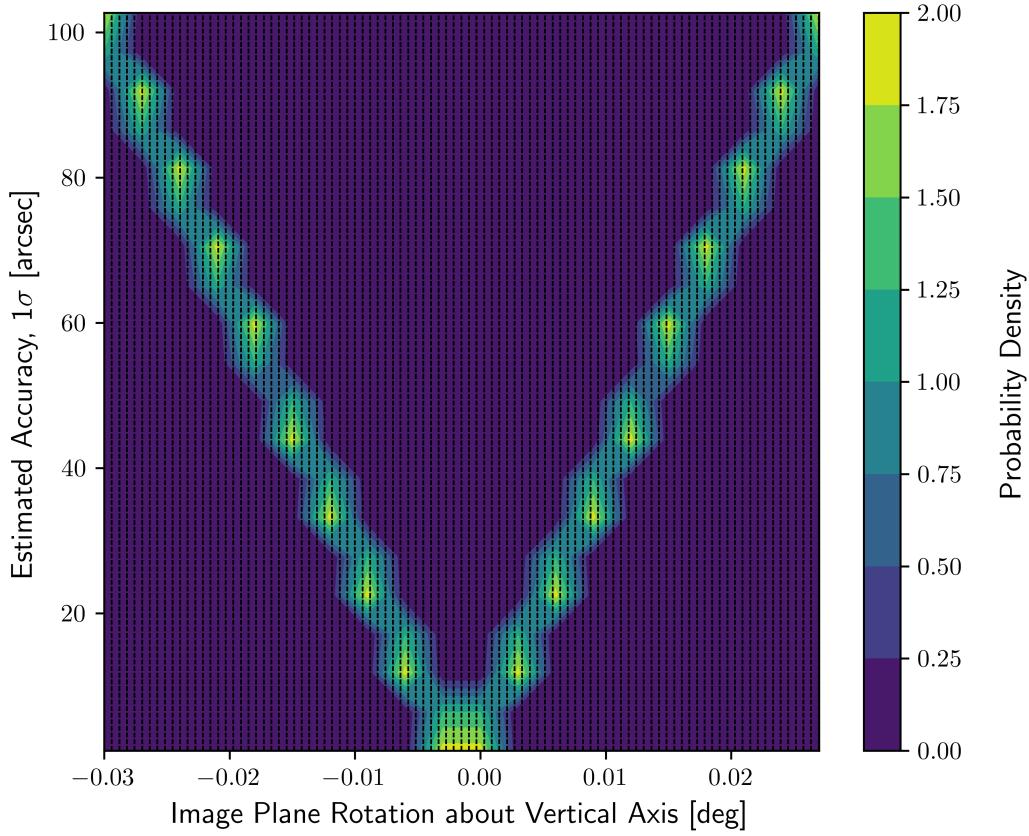


Figure 5.12: Sensitivity of accuracy with respect to image plane rotation about vertical axis

Based on the data presented in Figures 5.3 to 5.12, it can be deduced that the star tracker model is relatively robust to vertical image plane translation and rotation about its lateral axes. However, the attitude error is highly sensitive to lateral image plane translation, causing a constant bias in the measurement, and rotation about its vertical axis, inducing an artificial roll measurement in the system.

Centroiding Error

Aside from hardware aberrations, the star tracker is also subject to centroiding errors. Similar to the analysis performed for hardware aberration, Figure 5.13 presents the distribution of attitude estimate error when only the centroiding error is examined

within the star tracker model; i.e., the hardware is assumed to be ideal. Interestingly, centroiding accuracy does not lend itself to being a major contributor to the overall star tracker error presented in Figure 5.2. With an error of 0.401 ± 2.814 (1σ) arcseconds, the error for centroiding is relatively small compared to the other constituent parameters. It's also important to notice that centroiding should be characterized by a zero-mean distribution assuming the direction of the estimated centroid is uniformly random. Recall, in the model, centroiding error was computed in two steps: the magnitude of centroiding error was computed according to the magnitude of the star and the direction of error was derived from a parameter extracted from a uniform distribution. The process is reproduced in Equations 5.1 to 5.4 for reference.

$$\eta = \mathcal{N}(0, C(m)^2) \quad (5.1)$$

$$\theta = U(0, 2\pi) \quad (5.2)$$

$$\eta_x = \eta \cos \theta \quad (5.3)$$

$$\eta_y = \eta \sin \theta \quad (5.4)$$

where θ is the random direction parameter and $C(m)$ is the centroiding error computed from the star magnitude.

If the centroiding error has a consistent direction, the resulting distribution would fare similar to the distribution presented in Figure 5.5 caused by lateral image plane translation. The key difference between lateral image plane translation and centroiding error is that image plane translation affects each star uniformly, however, centroiding error affects each star independently of each other.

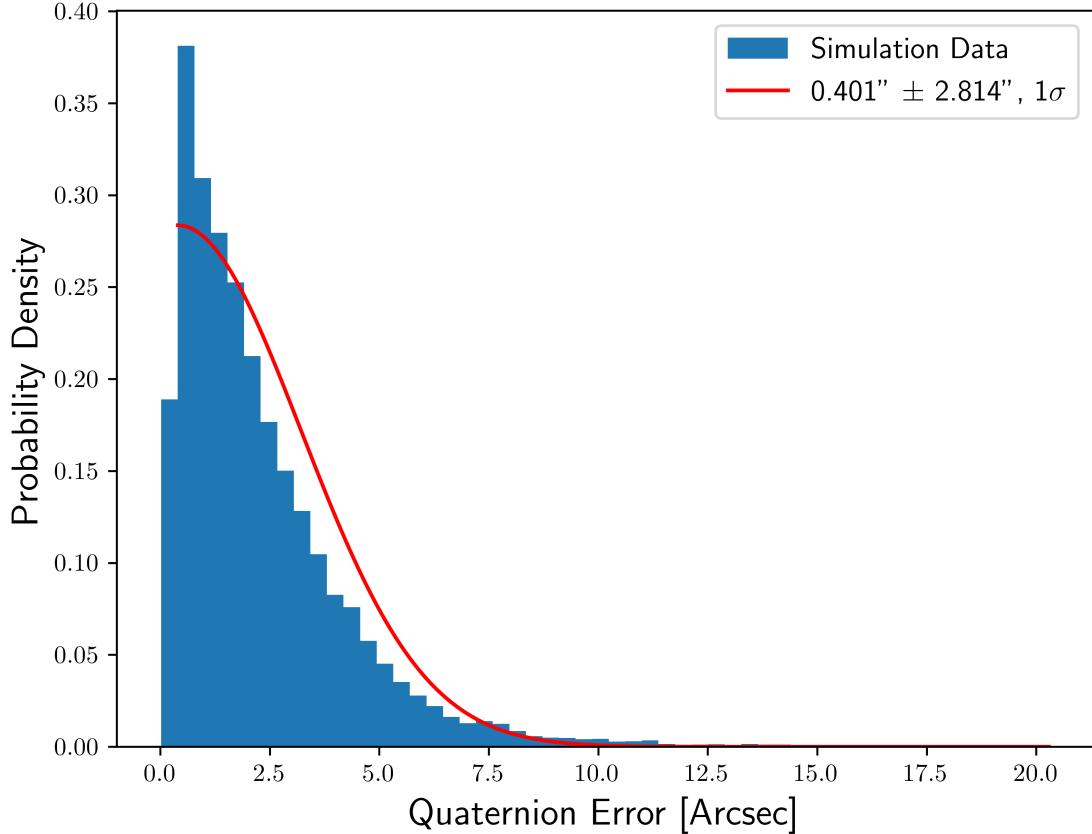


Figure 5.13: Attitude accuracy (0.401 ± 2.814 arcseconds, 1σ) versus centroid accuracy

To quantify the affects of centroiding error on star tracker accuracy, Figure 5.14 presents a sensitivity analysis which suggests that as centroiding error increases, the measurement quickly becomes uncertain and prone to large errors. This is a valid reflection of the underlying star tracker model; if the centroid is too far from its ideal position, then the projected star is too far from its original position. If each star in the image is subject to this large distortion, then the measured star scene would be too different from their corresponding representations in the ECI frame to generate a sound rotation. The contours of Figure 5.14 indicate that while the mean accuracy does not necessarily deviate, the uncertainty in attitude acquisition grows as centroiding accuracy decreases.

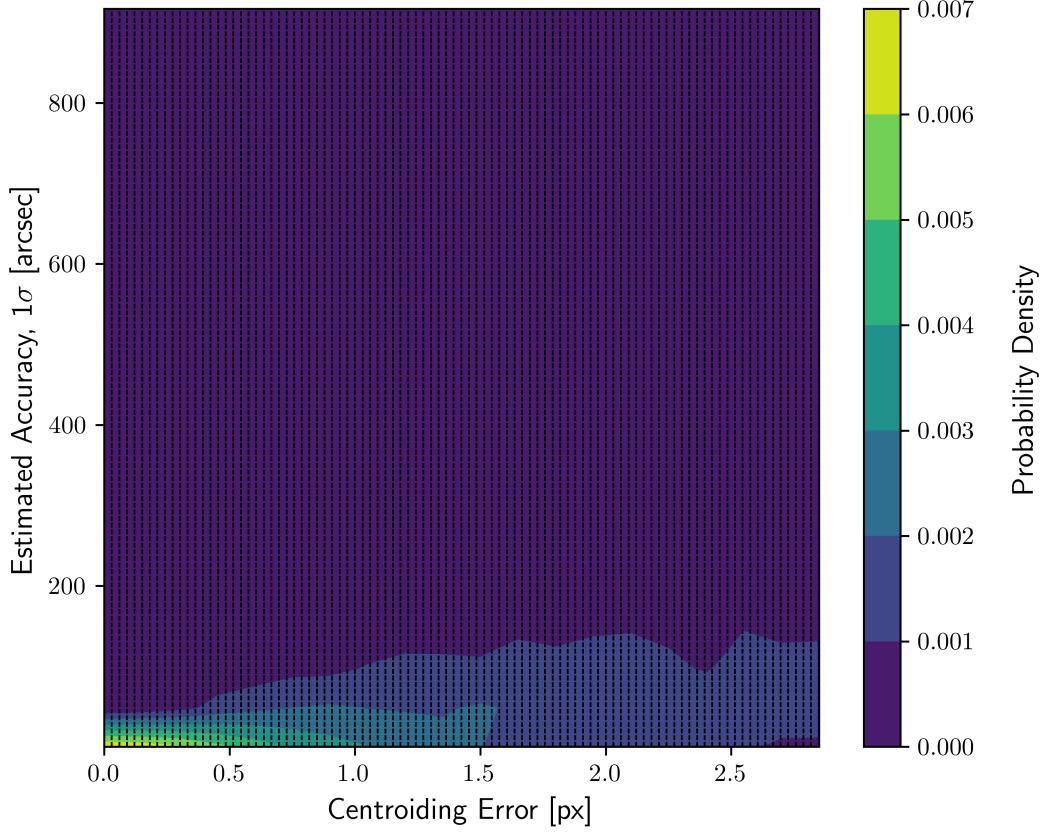


Figure 5.14: Sensitivity of accuracy with respect to centroid accuracy

Figure 5.13 examines the effects of centroiding accuracy based on the center of mass and star magnitude model presented by Liebe [11], however, alternative methods in determining star centroids also exist. In a paper from Yang et al. (2011), a least-squares vector-regression-based centroiding method was presented which was experimentally proven to improve centroiding accuracy from 0.06 pixels to 6e-5 pixels [37]. By replacing the center of mass model with the vector-regression based centroiding technique, the accuracy of the model star tracker significantly improves; Figure 5.15 presents the distribution. With an error distribution of 0.001 ± 0.005 (1σ) arcseconds, the alternative method improves upon the center of mass approximation by over 99%, albeit at a large computational cost. While not necessarily ideal for computation-limited star trackers, the alternative method is clearly superior

and would benefit any star tracker which is able to reconcile the resource allocation. Interestingly, the shape of the distribution is nearly identical to the center of mass technique which is to be expected as the methodology for simulating centroiding error was kept constant between the two.

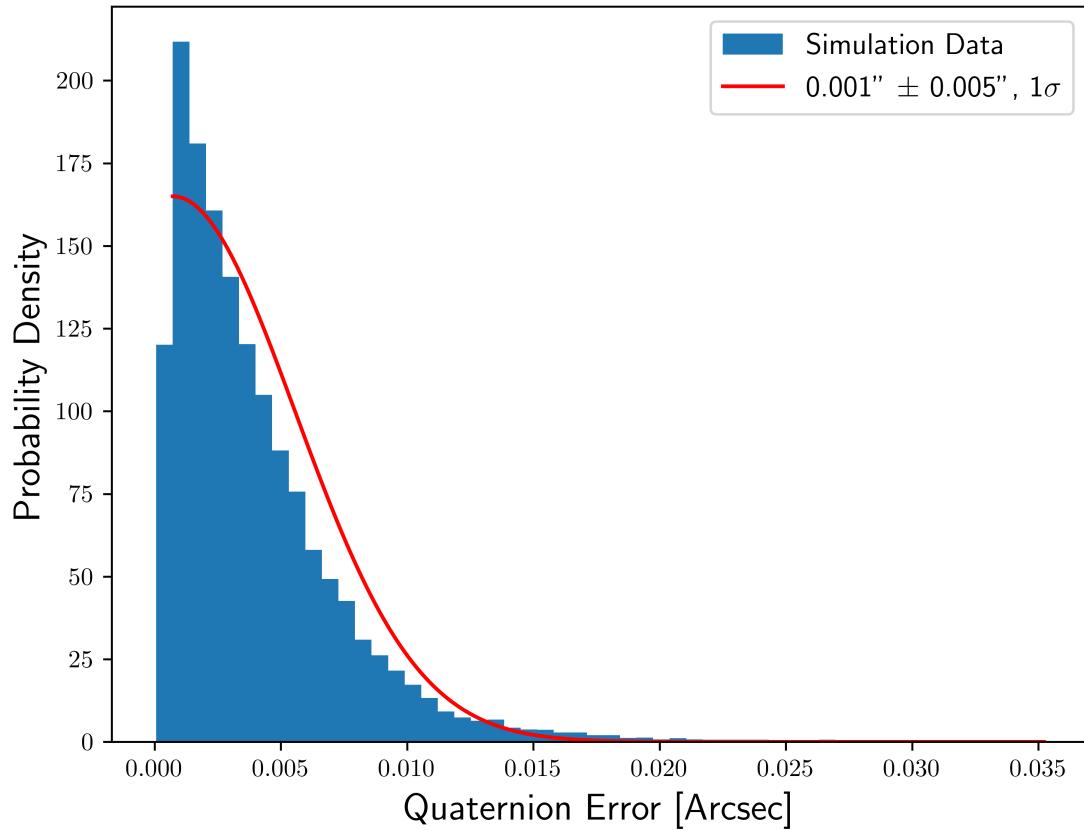


Figure 5.15: Attitude accuracy (0.001 ± 0.005 arcseconds, 1σ) versus least-squares vector regression centroid accuracy

Sensor Noise

While hardware aberration and centroiding noise will affect the incoming star vectors and cause error in the attitude estimate, the effects of sensor noise can entirely preclude the star tracker from capturing stars to begin with. Liebe presents a model for determining the photons collected from a star with a given apparent magnitude and Konnik presents a method for simulating image noise based on the incoming signal.

By combining the models with the sensor parameters outlined in Table 4.1, it was determined via simulation that the typical maximum magnitude visible by the sensor was 5.2 after applying the threshold recommended by Liebe which is fairly typical for a COTS CMOS sensor [11][24]. If a sensor is unable to discern enough stars from the celestial sphere during operations, it could fail the entry criteria to solving Wahba's Problem - to have at least 2 vector representations in each of the two reference frames. Figure 5.16 presents a plot outlining the rate of quantity-based star tracker measurement failures to the maximum magnitude a sensor can register. The model was generated by simulating the star tracker model but removing all stars from the scene with an apparent magnitude equal to or less than the target magnitude. As presented, it is clear that a sensor should at least be able to register stars with magnitude 5 to minimize this failure type. Figure 5.17 corroborates this supposition by indicating that as the maximum visible magnitude increases, the star tracker is less prone to failures when compared to star trackers with relatively low maximum visible magnitudes. It is evident that so long as the sensor is able to detect stars of magnitude 5 or lower, there is not a large influence on accuracy.

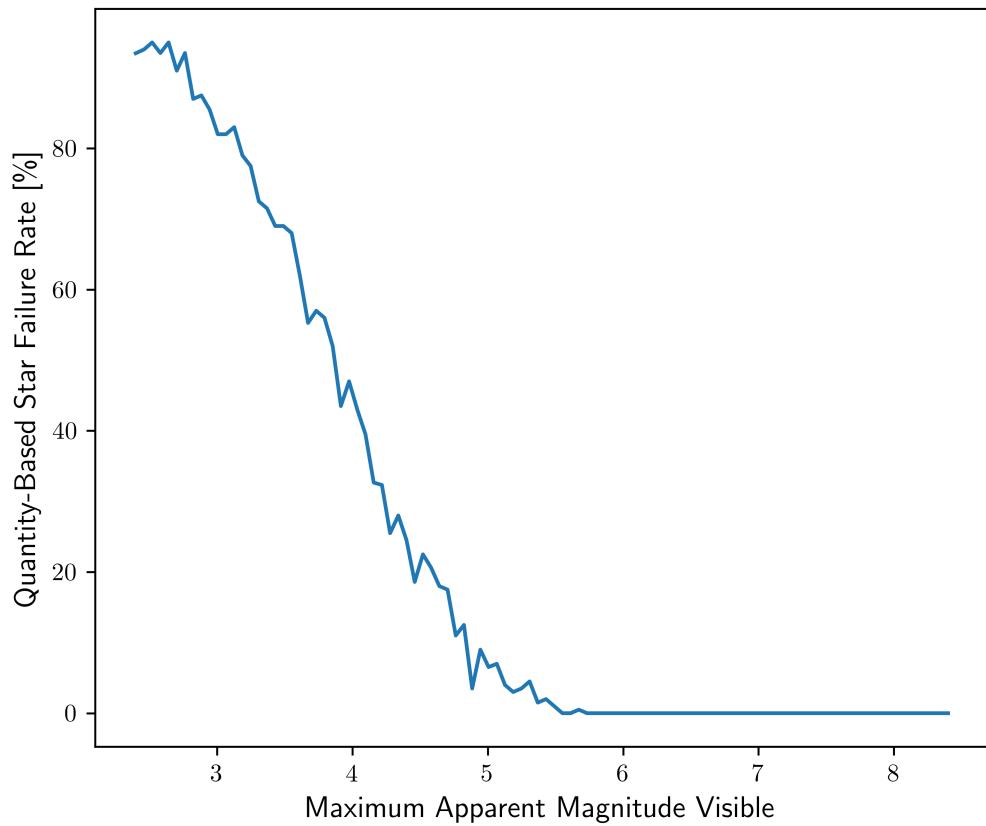


Figure 5.16: Star quantity-based failure rate with respect to maximum magnitude star discernable

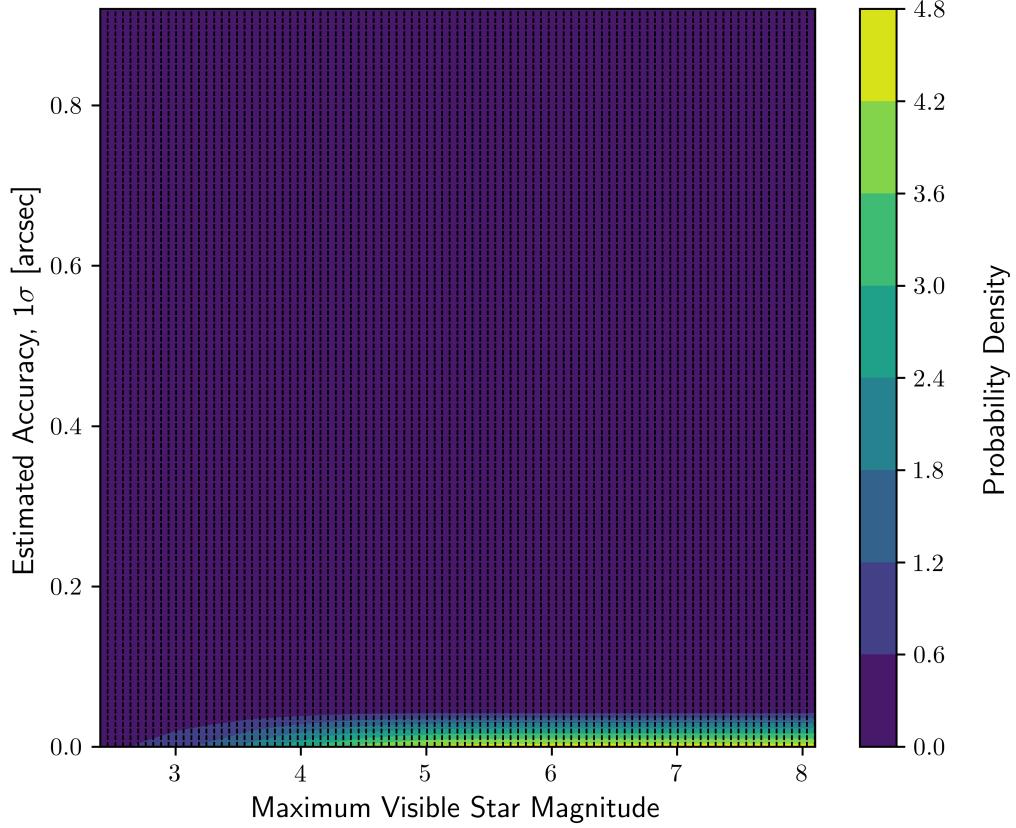


Figure 5.17: Sensitivity of accuracy with respect to maximum magnitude star discernable

Effects of the Space Environment

The results of the simulation presented above solely examined errors internal to the star tracker, however, in operations the star tracker is also subject to the thermal and radiation environment in LEO. The models presented below utilize the fully-parameterized star tracker presented in Table 4.1 and Figure 5.2 and interweaves the effects of the space environment to isolate their effects.

The effects of the thermal environment include the elongation/contraction of the focal length and the increased noise in the system. With an opto-thermal expansion coefficient probability density of $\mathcal{N}(7.9e-6, 3.9e-6^2)$ as presented by Jamieson et al. (1981), the focal length is projected to modulate only a few pixels [33]. Figure 5.18

presents the simulation data for a star tracker experiencing thermal swings. Interestingly, the effects of the thermal environment do not indicate a strong correlation with star tracker accuracy. Compared to the base star tracker distribution of 11.601 ± 28.376 arcseconds, the effects of the thermal environment increase the mean error to 12.515 arcseconds but decreases the standard deviation to 28.123 arcseconds. This data indicates that the thermal environment reduces uncertainty but increases the mean error - likely as a result of the noise in the image reducing the quality of the image and potentially removing stars from the image.

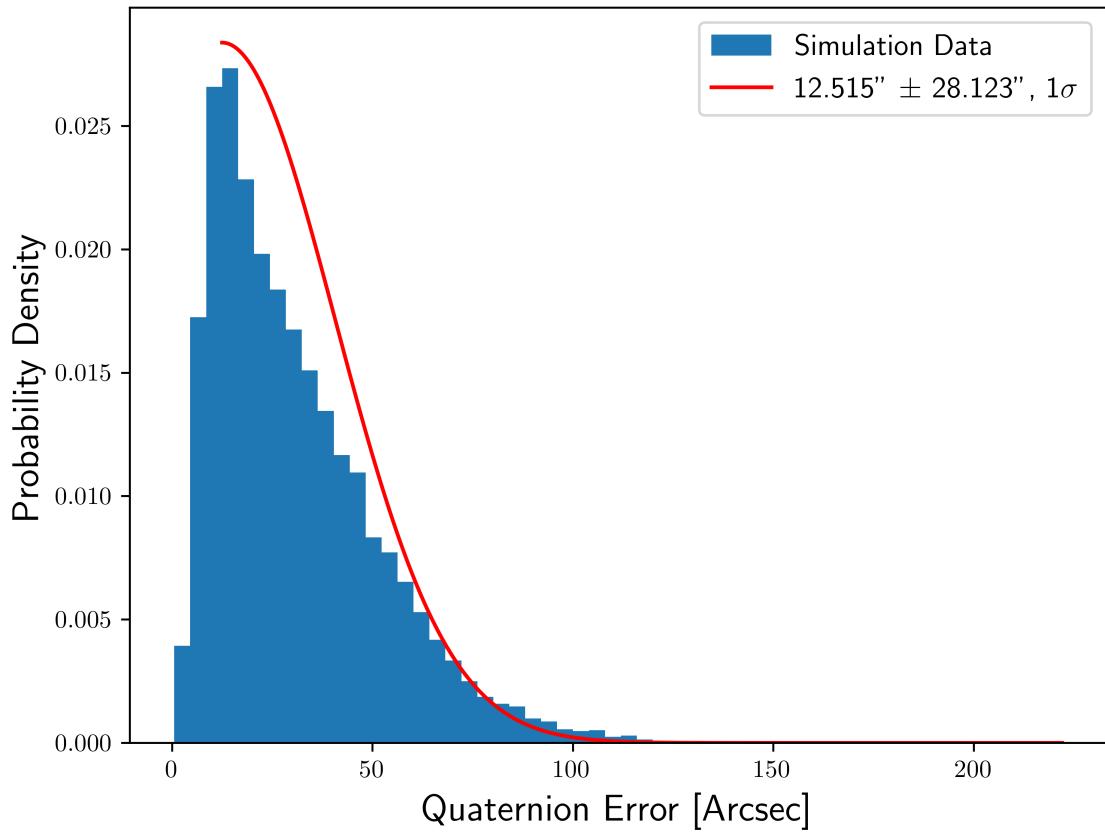


Figure 5.18: Attitude accuracy (12.515 ± 28.123 arcseconds, 1σ) with effects from the thermal environment

The radiation environment has a much stronger effect on the image noise. By applying the model for the increased dark current, increased fixed pattern noise, and decrease in overall responsivity, a simulated image with considerations of radiation

can be generated and the expected number of stars can be recomputed. Simulations indicate that the effects of radiation expectedly worsened sensor conditions. Previously with an approximate maximum magnitude threshold of 5.2, the effects post-ionization reduced the mean maximum magnitude to 5.11. Accounting for radiation, the resulting sensor characteristics remain within the window to minimize star quantity-based failures as given by Figure 5.16. The effects of radiation are presented in Figure 5.19 and indicate the degradation of the star tracker fidelity as it becomes exposed to radiation. Unlike the effects of the thermal environment, the radiation environment decreased the mean error compared to the base star tracker from 11.601 arcseconds to 11.206 arcseconds, but increased the uncertainty from 28.376 to 28.856 arcseconds.

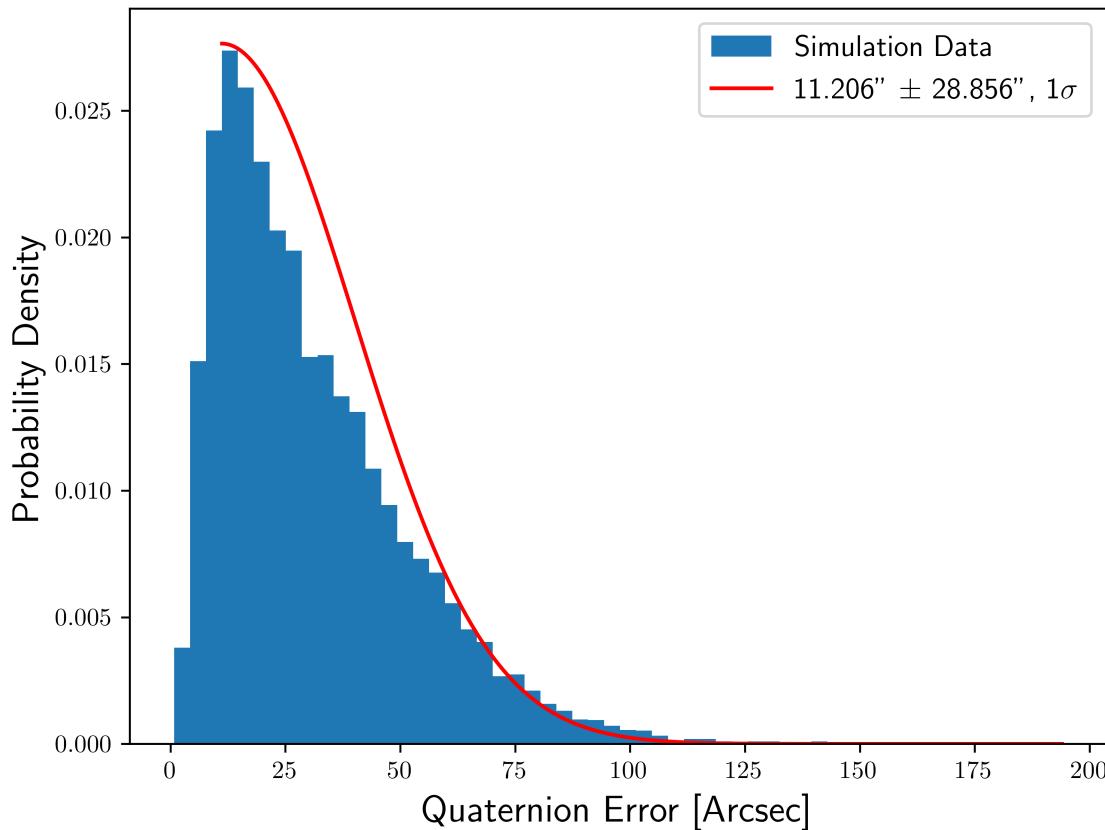


Figure 5.19: Attitude accuracy (11.206 ± 28.856 arcseconds, 1σ) with effects from the radiation environment

The effects of the thermal environment and radiation environment, interestingly, do not contribute significant error in the star tracker model compared to the internal parameters. To examine how different improvements affect the star tracker, a few special cases were compiled and compared to the base star tracker model.

5.3 Special Case Analysis

As the analysis of the star tracker model suggests that internal parameters are the true cause for large errors, it becomes crucial to consider a set of special cases. These encompass various calibration scenarios and different centroiding implementations, each having a potential to significantly influence the model's outcomes. Each model analyzed builds off the base star tracker parameterized by Table 4.1. The models include a complete hardware calibration, i.e., no hardware aberrations, calibration of the image plane translation and rotation separately, and an examination of the improved centroiding technique embedded within the base star tracker. Through considering these fringe cases, a more comprehensive understanding of the star tracker model's performance under diverse conditions can be determined.

Because image plane translation and rotation are likely caused by manufacturing and assembly imperfections, they are likely to remain constant throughout the lifetime of the star tracker. As a result, a total calibration can be achieved to effectively remove any hardware aberration [29]. Figure 5.20 presents the error distribution for a star tracker without any hardware aberration, but still subject to centroiding errors and sensor noise. It is clear that this avenue greatly improves upon the base star tracker. By completely eliminating hardware errors, the star tracker accuracy improves from 11.601 ± 28.376 arcseconds to 0.823 ± 3.329 arcseconds (1σ).

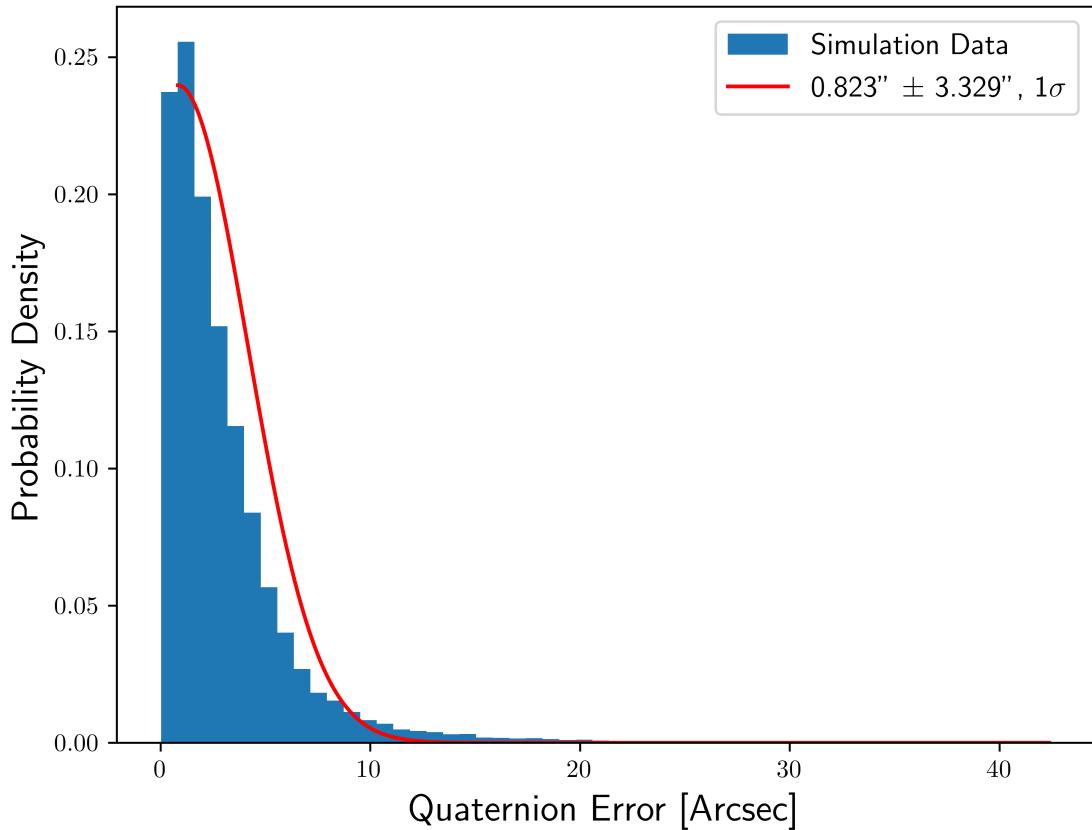


Figure 5.20: Attitude accuracy (0.823 ± 3.329 arcseconds, 1σ) of fully-calibrated star tracker model

To decompose the effects of calibration further, Figures 5.21 and 5.22 present the error distribution of a star tracker which is totally calibrated from image plane translation and image plane rotation, respectively, but still subject to centroid error and image noise. As expected these error models are very similar to individual counterparts presented in Figures 5.3 and 5.4, respectively, and further solidify the notion that accuracy is highly sensitive to hardware aberrations. By calibrating the star tracker for image plane translation, the bias in the measurement is nearly entirely eliminated, although at a cost of an increase in uncertainty. Similarly, by calibrating the star tracker for image plane rotation, the large uncertainty is reduced greatly from 28.376 arcseconds to 5.089 arcseconds (1σ), although at the cost of introducing a constant bias in an unknown direction.

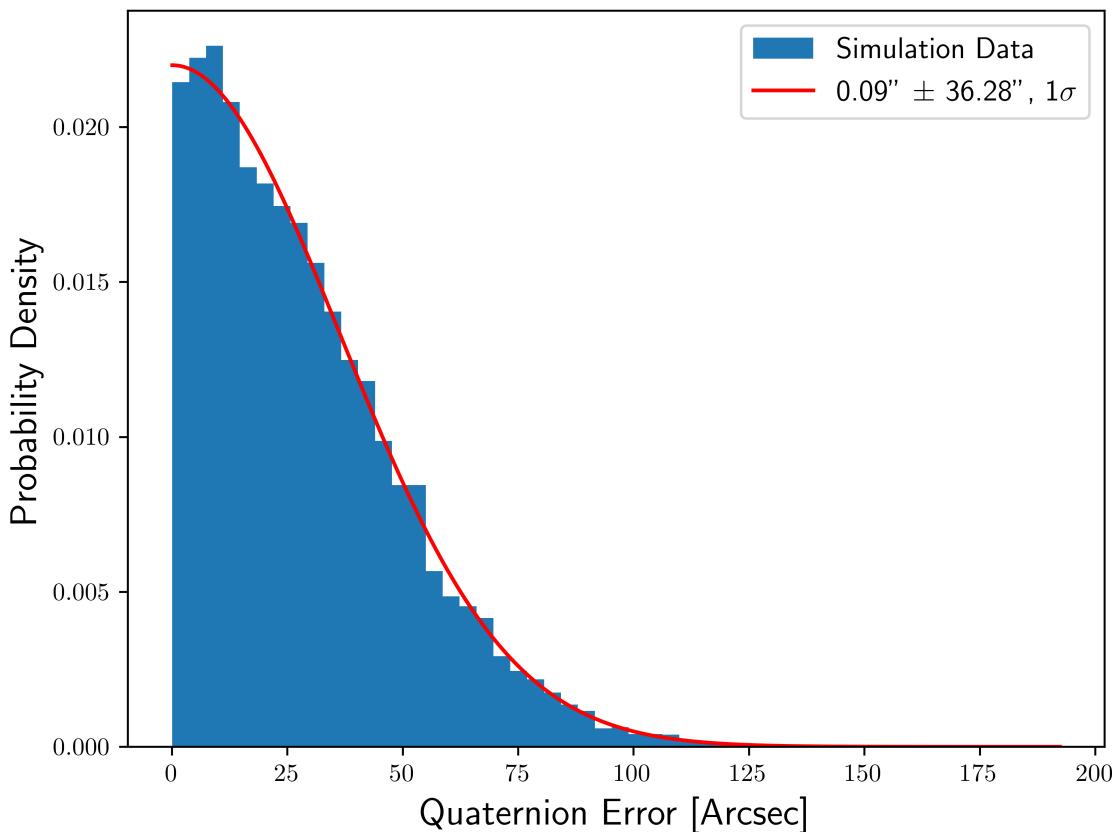


Figure 5.21: Attitude accuracy (0.090 ± 36.280 arcseconds, 1σ) of translation-calibrated star tracker model

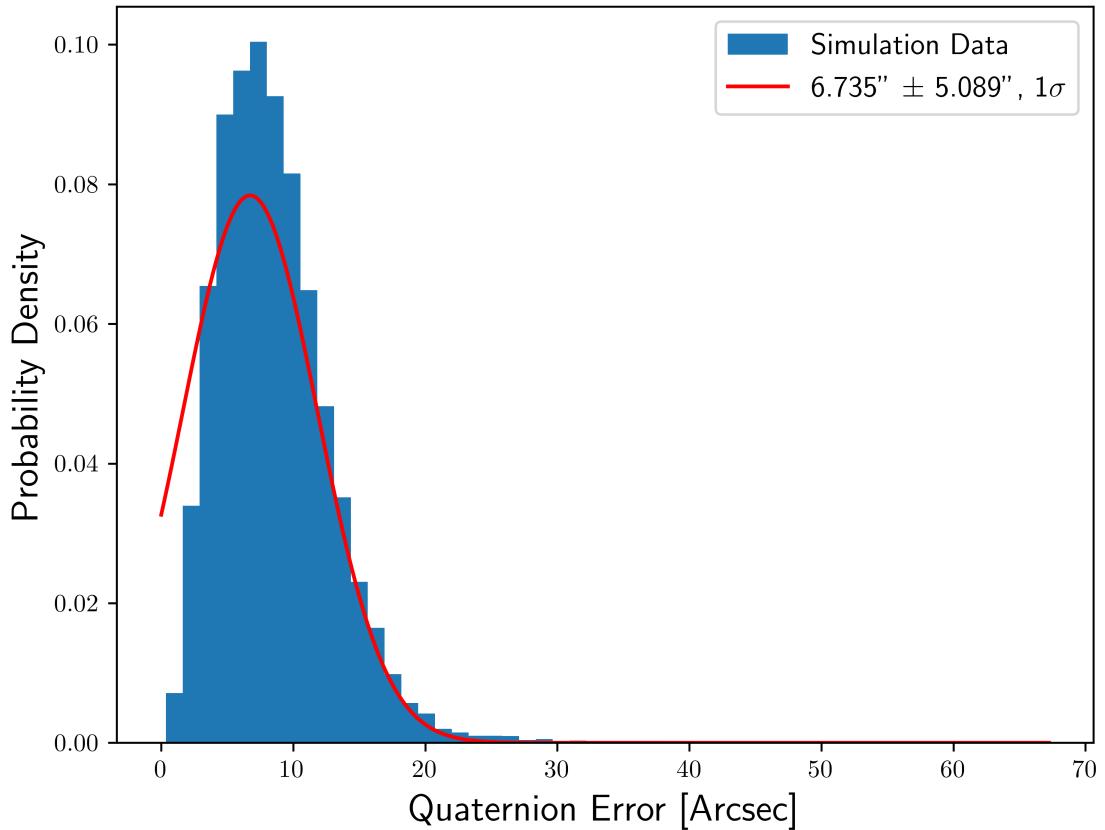


Figure 5.22: Attitude accuracy (6.735 ± 5.089 arcseconds, 1σ) of rotation-calibrated star tracker model

Another potential improvement to the star tracker is to implement the aforementioned vector-regression centroiding technique. The effects of applying the method is shown to have an effect on a star tracker with non-ideal hardware. With an improvement of 2.177 arcseconds in the bias but an increase of 1.349 arcseconds (1σ) in the uncertainty, the increased centroiding fidelity is not overshadowed by the defects in the star tracker hardware and can be especially useful when hardware calibration techniques or resources are not available. Figure 5.23 presents the accuracy distribution for a star tracker with non-ideal hardware and the least-squares vector regression centroiding method.

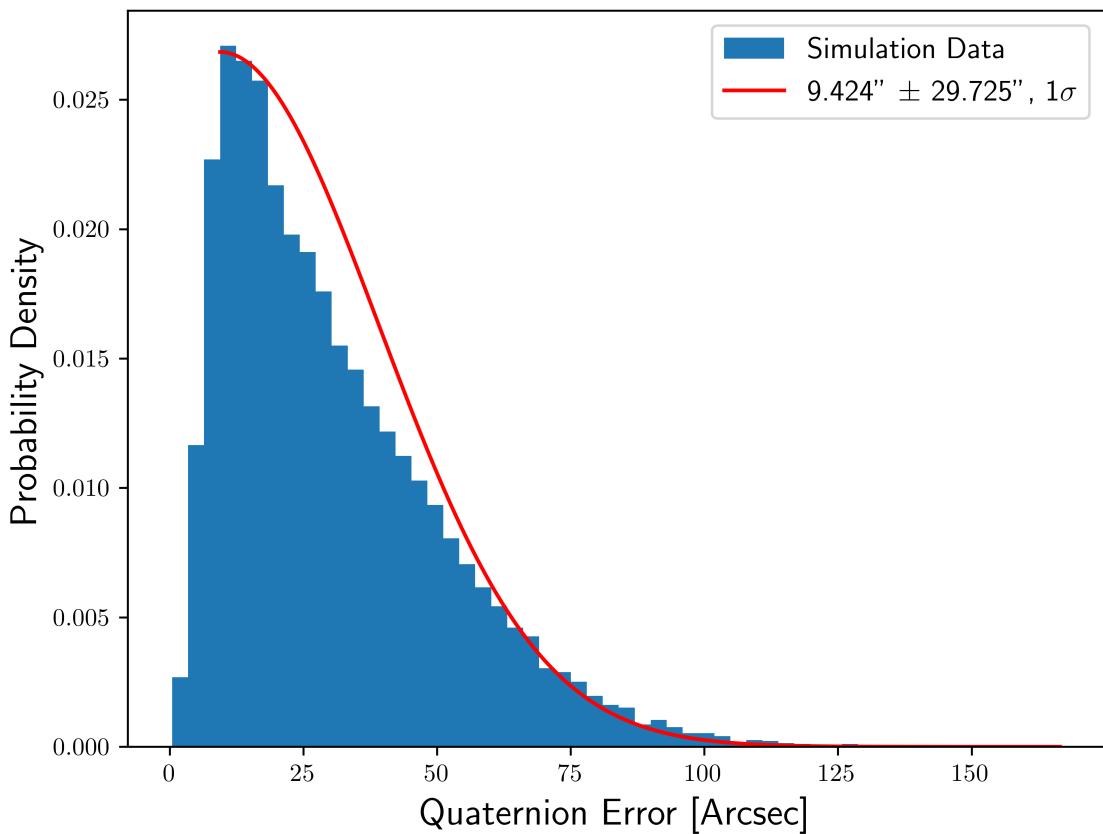


Figure 5.23: Attitude accuracy (9.424 ± 29.725 arcseconds, 1σ) of improved-centroiding star tracker model

Chapter 6

RESULTS AND CONCLUSION

6.1 Conclusion

This thesis has presented an exhaustive investigation into error propagation in star tracker systems. A comprehensive measurement process model was developed from first principles, which allowed for a meticulous examination of the sources and impacts of errors in star trackers. The primary sources of errors examined include distortions in hardware, image noise, centroiding accuracy, and the thermal and radiation environment in LEO.

The development of the star tracker model was undertaken in an incremental fashion, providing opportunities to continually validate and adjust the design to closely mirror reality. Each stage of the development process was approached with deliberation and scrutiny to ensure the model's fidelity and accuracy. The initial design incorporated fundamental principles of astrodynamics and celestial navigation, forming the core of the model. Gradually, more complex elements, such as aberrations, distortions, and noise, were layered in to refine the model and reflect real-world conditions. Design decisions throughout the development process were made judiciously, guided by two main considerations. The goal was to preserve the model's validity, ensuring that every new component or refinement was grounded in empirical evidence and theoretical principles. Maintaining a close resemblance to real-world conditions was a priority. This often necessitated a delicate balance between complexity and simplicity, as overly complex models can become unwieldy and lose their practical applicability, while overly simple models may not accurately reflect the nuances of reality. Through this iterative and meticulous approach, a star tracker model was

developed that not only remains valid and verifiable but also faithfully replicates the complexities of real-world star tracking with space to enable future improvements to further improve the realism.

This measurement process model serves dual purposes. Firstly, it can be employed to effectively characterize star trackers during the developmental phases, thereby providing engineers with valuable insights into the performance of these systems early on. Secondly, it can function as a high-fidelity software model suitable for integration into broader spacecraft simulations.

It was observed that the most significant contributor to error in accuracy was image plane rotation, though image plane translation consistently induced a bias. Fortunately, both these errors can be substantially mitigated through calibration during the development phase [30][29]. In scenarios where calibration hardware may not be accessible, augmenting the accuracy of centroiding provides a substantial boost to overall star tracker accuracy. Other parameters such as vertical image plane translation or rotation of the sensor about the lateral axes proved to not cause much error in the model. Similarly, the effects from the space environment, specifically the thermal and radiation environment, were also simulated and shown to not have much of an effect on star tracker accuracy. The sensitivity analysis further amplifies the practical utility of our model. It elucidates the influence of various inputs on the overall star tracker accuracy, thereby facilitating the prioritization of developmental tasks.

To summarize, this thesis provides a solid foundation for future work in the realm of star trackers, laying the groundwork for improved error prediction and mitigation strategies. By helping prioritize development tasks and facilitating more accurate simulations, the star tracker measurement process model promises to play a significant role in enhancing the precision and reliability of star tracking systems in future space missions.

6.2 Future Work

The research findings and foundational groundwork established in this thesis have uncovered some potential trajectories for subsequent investigations that can broadly be broken down into two categories: simulation framework and core model improvements.

Simulation Framework

Simulation framework revolves around the infrastructure surrounding the core star tracker model. Aside from model optimizations to improve performance, one suggested extension involves obtaining experimental data to validate the proposed star tracker model. It is critical to verify that the theoretical constructs and simulation outcomes closely align with empirical observations and measurements. By aligning simulation results to known and experimental star tracker data, the model can provide information about unknown star tracker states with a greater degree of accuracy and precision.

For future research, it would be beneficial to focus on constructing a more robust and deterministic simulation architecture, instead of relying on a stopping criterion. This new approach would move away from dependence on error tolerances, potentially resulting in a simulation model that is more predictable and stable. A well-founded simulation architecture could provide more consistent results, reduce the need for large amounts of data, and ultimately enhance the accuracy and reliability of the simulations. Additionally, such an architectural overhaul could offer avenues for introducing more complex parameters, leading to more nuanced simulations.

Core Model Improvements

Several properties of the star tracker were not considered for this model for a variety of reasons. Developing additional modules and relations and increasing the complexity would ground the the model further in reality and provide more tunable parameters to more accurately simulate a star tracker.

The significance of lens distortion in the precision of star trackers cannot be understated. Lens distortion can cause additional positioning error of the star on the focal array, worsening the effects of hardware aberration. Although various calibration methods are designed to minimize lens distortion, a comprehensive investigation can yield deeper insights and draw the star tracker's operation closer to real-world conditions. In addition to considering lens distortion, there is a crucial need to undertake a thorough examination of the influence of manufacturing defects unique to star tracker equipment. Gaining insights into how these defects affect performance will allow a more accurate portrayal of the impacts of hardware aberrations. Furthermore, calibration techniques do not guarantee the removal of all hardware defects, but can greatly reduce their effects. An analysis of various calibration techniques and the improvements brought with them can provide great insight into various avenues of star tracker improvement.

Future research should delve into the potential skewing or failure of measurements due to the identification technique. This thesis does not examine the effects of identification as the Pyramid Algorithm was discussed to be highly accurate and robust. However, not all star trackers employ the Pyramid Algorithm; a survey of the effects of identification coupled with sensor noise and star-positioning error can further improve the validity of the star tracker model. By considering effects such as charged particles creating false stars in the image or the abundance of noise reducing the number of discernable stars, additional insight and troubleshooting capabilities can

be introduced. Another emerging properties of star trackers is the implementation of recursive identification. Star trackers are commonly paired with gyroscopes or some other angular sensor to estimate the attitude with some relatively large uncertainty before exactly determining it via the star tracker measurement process. The sensor fusion enables the star tracker to search a smaller space and reduce the chances of incorrect attitude estimations. The influence of the supporting infrastructure on the overall accuracy and reliability of the star tracker model should be examined to illustrate how sensor fusion techniques can improve attitude determination.

Continued research and exploration in these areas will not only further enhance our understanding of error propagation in star trackers but will also contribute to the development of more accurate, reliable, and robust star tracking systems. This will ultimately bolster our capabilities in space navigation and enable spacecraft to accurately determine the attitude encoded within the stars.

BIBLIOGRAPHY

- [1] California Polytechnic State University. Cubesat design specification. Rev. 14.1, California Polytechnic State University, 2022.
- [2] Cal Poly CubeSat Laboratory. Cal poly cubesat laboratory, 2023.
- [3] NASA. Cubesats: Tiny payloads, huge benefits for space research, 2023.
Accessed: 2023-05-21.
- [4] Peter Grenfell, Alexa Aguilar, Kerri Cahoy, and Michael Long. Pointing, acquisition, and tracking for small satellite laser communications.
Proceedings of the AIAA/USU, 2018.
- [5] Alberto Carrasco-Casado, Abhijit Biswas, Renny Fields, Brian Grefenstette, Fiona Harrison, Suzana Sburlan, and Morio Toyoshima. Optical communication on cubesats — enabling the next era in space science. In *2017 IEEE International Conference on Space Optical Systems and Applications (ICSOS)*, pages 46–52, 2017.
- [6] Mohamad Dol Bahar, Mohd Effandi, Mohd Khair Bin Hassan, Norhizam Hamzah, Ahmad Sabirin Arshad, Xandri C. Farr, Lourens Visagie, and Willem H. Steyn. Modular cmos horizon sensor for small satellite attitude determination and control subsystem. 2006.
- [7] CubeSatShop. nanossoc-a60 datasheet. Accessed: 2023-05-21.
- [8] CubeSatShop. arcsec sagitta star tracker, 2023. Accessed: 2023-05-21.
- [9] RocketLab. St-16rt2 datasheet, 2023. Accessed: 2023-05-21.

- [10] Josh Grace, Luca Merlo Paula Soares, Trevor Loe, and John Bellardo. *A Low Cost Star Tracker for CubeSat Missions*.
- [11] C.C. Liebe. Accuracy performance of star trackers - a tutorial. *IEEE Transactions on Aerospace and Electronic Systems*, 38(2):587–599, 2002.
- [12] Anton H.J. de Ruiter. *Spacecraft Dynamics and Control: An Introduction*. Wiley, 2013.
- [13] Ahmad Bani Younes and D. Mortari. Derivation of all attitude error governing equations for attitude filtering and control. *Sensors*, 19, 10 2019.
- [14] Grace Wahba. Problem 65-1: A least squares estimate of satellite attitude. *SIAM Review*, 7(3):409–409, 1965.
- [15] Landis Markley. Attitude determination using vector observations and the singular value decomposition. *J. Astronaut. Sci.*, 38, 11 1987.
- [16] 1997 Esa. VizieR Online Data Catalog: The Hipparcos and Tycho Catalogues (ESA 1997). *VizieR Online Data Catalog*, page I/239, February 1997.
- [17] D. Hoffleit and Jr. Warren, W. H. VizieR Online Data Catalog: Bright Star Catalogue, 5th Revised Ed. (Hoffleit+, 1991). *VizieR Online Data Catalog*, page V/50, November 1995.
- [18] Yongyong Li, Xinguo Wei, Jian Li, and Gangyi Wang. Error correction of rolling shutter effect for star sensor based on angular distance invariance using single frame star image. *IEEE Transactions on Instrumentation and Measurement*, 71:1–13, 2022.
- [19] David A. Forsyth and Jean Ponce. *Computer Vision - A Modern Approach, Second Edition*. Pitman, 2012.

- [20] Daniele Mortari, Malak A. Samaan, Christian Bruccoleri, and John L. Junkins. The pyramid star identification technique. *NAVIGATION*, 51(3):171–183, 2004.
- [21] T.S. Kelso. TLE data: Cubesats (current as of 2023 may 17), 05 2023.
- [22] National Aeronautics and Space Administration. Space radiation effects on electronic components in low-earth orbit, 02 1999.
- [23] G.R. Hopkinson. Radiation effects in a cmos active pixel sensor. *IEEE Transactions on Nuclear Science*, 47(6):2480–2484, 2000.
- [24] Mikhail Konnik and James Welsh. High-level numerical simulations of noise in ccd and cmos photosensors: review and tutorial, 2014.
- [25] Joseph Edward Rushton. *Radiation Damage in CMOS Image Sensors for Space Applications*. Open University (United Kingdom), 2018.
- [26] Hui Jia, Jiankun Yang, Xiujuan Li, Juncai Yang, MengFei Yang, YiWu Liu, and YunCai Hao. Systematic error analysis and compensation for high accuracy star centroid estimation of star tracker. *Science China Technological Sciences*, 53:3145–3152, 11 2010.
- [27] A. Read Eisenman and C.C. Liebe. The advancing state-of-the-art in second generation star trackers. In *1998 IEEE Aerospace Conference Proceedings (Cat. No.98TH8339)*, volume 1, pages 111–118 vol.1, 1998.
- [28] Michel Dekking. *A Modern Introduction to Probability and Statistics: Understanding Why and How*. Longon: Springer, 2005.
- [29] HaiBo Liu, Xiujuan Li, Ji-Chun Tan, Jian-Kun Yang, Jun Yang, De-Zhi Su, and Hui Jia. Novel approach for laboratory calibration of star tracker. *Optical Engineering*, 49:3601–, 07 2010.

- [30] Ting Sun, Fei Xing, and Zheng You. Optical system error analysis and calibration method of high-accuracy star trackers. *Sensors (Basel, Switzerland)*, 13:4598–623, 04 2013.
- [31] Peter Harris and Maurice Cox. On a monte carlo method for measurement uncertainty evaluation and its implementation. *Metrologia*, 51:S176, 07 2014.
- [32] G.E. Healey and R. Kondepudy. Radiometric ccd camera calibration and noise estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):267–276, 1994.
- [33] Thomas H Jamieson. Thermal effects in optical systems. *Optical Engineering*, 20(2):156–160, 1981.
- [34] Alejandro Garzón and Yovani Villanueva. Thermal analysis of satellite libertad 2: a guide to cubesat temperature prediction. *Journal of Aerospace Technology and Management*, 10, 11 2018.
- [35] Vincent Goiffon. *Radiation Effects on CMOS Active Pixel Image Sensors*, pages 295–332. 11 2015.
- [36] Michail Tsagris, Christina Beneki, and Hossein Hassani. On the folded normal distribution. *Mathematics*, 2(1):12–28, 2014. Copyright - Copyright MDPI AG 2014; Last updated - 2014-08-11.
- [37] Jun Yang, Bin Liang, Tao Zhang, and Jingyan Song. A novel systematic error compensation algorithm based on least squares support vector regression for star sensor image centroid estimation. *Sensors*, 11(8):7341–7363, 2011.

- [38] C Federico, AM Di Lellis, S Fonte, C Pauselli, G Reitz, and R Beaujean.
Exomars iras (dose) radiation measurements. *Mem. Soc. Astron. Ital.*(*Suppl. 11*), 2007.

APPENDICES

Appendix A

HARDWARE ABERRATION MODEL

$$\mathbf{r}_{F/\gamma,\gamma} = \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix} \quad \mathbf{r}_{\gamma/\pi,\gamma} = \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \end{bmatrix} \quad (\text{A.1})$$

$$\mathbf{r}_{F/\pi,\gamma} = \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ F + \epsilon_z \end{bmatrix} \quad (\text{A.2})$$

$$\mathbf{C}_{\gamma\pi} = \vec{\mathcal{F}}_\gamma \vec{\mathcal{F}}_\pi^T \quad (\text{A.3})$$

$$= \mathbf{R}_x(\phi_x) \mathbf{R}_y(\theta_y) \mathbf{R}_z(\psi_z) \quad (\text{A.4})$$

$$\mathbf{r}_{F/\pi,\pi} = \mathbf{C}_{\gamma\pi} \mathbf{r}_{F/\pi,\gamma} \quad (\text{A.5})$$

$$\mathbf{S} = \mathbf{r}_{S/F,\gamma} \quad (\text{A.6})$$

$$\mathbf{r}_{S/\gamma,\gamma} = \mathbf{r}_{F/\gamma,\gamma} + \mathbf{S} \quad (\text{A.7})$$

$$\mathbf{r}_{S/\pi,\gamma} = \mathbf{r}_{F/\gamma,\gamma} + \mathbf{r}_{\gamma/\pi,\gamma} + \mathbf{S} \quad (\text{A.8})$$

$$\mathbf{r}_{S/\pi,\gamma} = \mathbf{r}_{F/\pi,\gamma} + \mathbf{S} \quad (\text{A.9})$$

$$\mathbf{r}_{S/\pi,\pi} = \mathbf{C}_{\gamma\pi} (\mathbf{r}_{F/\pi,\gamma} + \mathbf{S}) \quad (\text{A.10})$$

$$\mathbf{r}_{S/\pi,\pi} = \mathbf{r}_{F/\pi,\pi} + \mathbf{C}_{\gamma\pi} \mathbf{S} \quad (\text{A.11})$$

$$\begin{bmatrix} x_\pi \\ y_\pi \\ z_\pi \end{bmatrix} = \mathbf{r}_{F/\pi,\pi} + \lambda \mathbf{C}_{\gamma\pi} \mathbf{S} \quad (\text{A.12})$$

$$\begin{bmatrix} x_\pi^* \\ y_\pi^* \\ z_\pi^* = 0 \end{bmatrix} = \mathbf{r}_{F/\pi, \pi} + \lambda^* \mathbf{C}_{\gamma\pi} \mathbf{S} \quad (\text{A.13})$$

$$\lambda^* = \frac{- (\mathbf{r}_{F/\pi, \pi})_z}{(\mathbf{C}_{\gamma\pi} \mathbf{S})_z} \quad (\text{A.14})$$

$$\hat{\mathbf{S}} = \frac{1}{\sqrt{x_\pi^{*2} + y_\pi^{*2} + F^2}} \begin{bmatrix} -x_\pi^* \\ -y_\pi^* \\ F \end{bmatrix} \quad (\text{A.15})$$

Appendix B

CONNECTED COMPONENT ALGORITHM

Algorithm 6 Connected Component Algorithm

Require: Image

```

label = 0
Bmin ← Brightness Threshold
ROI ← ∅                                ▷ Structure mapping pixels to their label
w ← Image Width
h ← Image Height

function CHECKNEIGHBORS(i, j)
  for x = -1 to 1 do
    for y = -1 to 1 do
      if x = 0 and y = 0 then skip          ▷ skips pixel in question
      end if
      Pxy ← Image[i + x, j + y]
      if Pxy in ROI then
        return ROI[Pxy]    ▷ Returns label associated with neighboring pixel
      end if
    end for
  end for
  label = label + 1    ▷ Creates new label if no neighboring pixel is registered
  return label          ▷ Return identifying label
end function

function FINDREGIONSOFINTEREST(Image)
  for i = 0 to w do
    for j = 0 to h do
      Pij ← Image[i, j]                  ▷ Extract pixel data at location (i, j)
      if Pij ≥ Bmin then
        pixelLabel ← CheckNeighbors(i, j)
        ROI[Pij] = pixelLabel    ▷ Create mapping between label and pixel
      end if
    end for
  end for
end function
  
```
