2016

Manual Optical Attitude Re-initialization of a Crew Vehicle in Space Using Bias Corrected Gyro Data

Christopher J. Gioia

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Recommended Citation

Gioia, Christopher J., "Manual Optical Attitude Re-initialization of a Crew Vehicle in Space Using Bias Corrected Gyro Data" (2016). *Graduate Theses, Dissertations, and Problem Reports*. 5676.
<https://researchrepository.wvu.edu/etd/5676>

This Dissertation is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Dissertation in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Dissertation has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

Manual Optical Attitude Re-initialization of a Crew Vehicle in Space Using Bias Corrected Gyro Data

Christopher J. Gioia

Dissertation submitted to
the Benjamin M. Statler College Of Engineering And Mineral Resources
at West Virginia University
in partial fulfillment of the requirements for the degree of
Doctorate of Philosophy in Mechanical Engineering

John A. Christian, Chair Ph.D.
Mario Perhinschi, Ph.D.
Patrick Browning, Ph.D.
Jason Gross, Ph.D.
Andrew Lowery, Ph.D.

Department of Mechanical and Aerospace Engineering

Morgantown, West Virginia 2016

Keywords: Navigation, Attitude Estimation, Wahba Problem, Gyro Bias, Optimization,
Spacecraft, Human Space Exploration

Copyright © 2016 Christopher J. Gioia

ABSTRACT

Manual Optical Attitude Re-initialization of a Crew Vehicle in Space Using Bias Corrected Gyro Data

Christopher J. Gioia

NASA and other space agencies have shown interest in sending humans on missions beyond low Earth orbit. Proposed is an algorithm that estimates the attitude of a manned spacecraft using measured line-of-sight (LOS) vectors to stars and gyroscope measurements. The Manual Optical Attitude Re-initialization (MOAR) algorithm and corresponding device draw inspiration from existing technology from the Gemini, Apollo and Space Shuttle programs. The improvement over these devices is the capability of estimating gyro bias completely independent from re-initializing attitude. It may be applied to the lost-in-space problem, where the spacecraft's attitude is unknown.

In this work, a model was constructed that simulated gyro data using the Farrenkopf gyro model, and LOS measurements from a spotting scope were then computed from it. Using these simulated measurements, gyro bias was estimated by comparing measured interior star angles to those derived from a star catalog and then minimizing the difference using an optimization technique. Several optimization techniques were analyzed, and it was determined that the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm performed the best when combined with a grid search technique. Once estimated, the gyro bias was removed and attitude was determined by solving the Wahba Problem via the Singular Value Decomposition (SVD) approach. Several Monte Carlo simulations were performed that looked at different operating conditions for the MOAR algorithm. These included the effects of bias instability, using different constellations for data collection, sampling star measurements in different orders, and varying the time between measurements. A common method of estimating gyro bias and attitude in a Multiplicative Extended Kalman Filter (MEKF) was also explored and disproven for use in the MOAR algorithm.

A prototype was also constructed to validate the proposed concepts. It was built using a simple spotting scope, MEMS grade IMU, and a Raspberry Pi computer. It was mounted on a tripod, used to target stars with the scope and measure the rotation between them using the IMU. The raw measurements were then post-processed using the MOAR algorithm, and attitude estimates were determined. Two different constellations — the Big Dipper and Orion — were used for experimental data collection. The results suggest that the novel method of estimating gyro bias independently from attitude in this document is credible for use onboard a spacecraft.

To my family...

Acknowledgments

I would like to begin by thanking my adviser Dr. John Christian. Thank you for the opportunity to work on a NASA project, especially one with potential huge impact on future deep space missions. Thank you for your guidance, insight, feedback, advice, and instruction on proper use of the Oxford comma. I could not have asked for a better adviser throughout my PhD pursuit.

I would also like to thank the members of my graduate committee, Dr. Mario Perhinschi, Dr. Patrick Browning, Dr. Jason Gross, and Dr. Drew Lowery. Thank you for all of your advice on my research and feedback on my manuscript.

Thank you to Andrew Liounis and Dr. John O. Woods for helping me get my prototype coded up and functional. I would also like to thank Dr. Jay Wilhelm for not only providing me with guidance during my Master's pursuit, but continuing to provide me with guidance throughout my PhD pursuit and assistance with my prototype.

I would like to thank Kathryn VanDeveer, Ryan Watson, Jacob Hikes, Andrew and Natalie Liounis, and Andrew Rhodes for helping me collect data with the prototype. Without your help there was no way I would have been able to get everything up to the roof and accurately record data.

I would especially like to thank my family. To my parents Angelo and Maureen Gioia, thank you for supporting me throughout my life in everything that I do. I appreciate all of the love and support you've shown me throughout my life and especially my education. Thank you to my sister Sarah for always being there for me as well. Finally, I would like to thank my girlfriend Kathryn VanDeveer for being there and supporting me in everything that I do.

Sincerely,

Christopher J. Gioia

Contents

Abstract	ii
Acknowledgments	iv
Nomenclature	xi
1 Introduction	1
1.1 Motivation	2
1.2 Research Objectives	4
1.2.1 Objective 1: Numerical Simulation of the System	4
1.2.2 Objective 2: Removal of Gyro Bias	5
1.2.3 Objective 3: Determine Importance of Accuracy vs. Time of Bias Estimate	5
1.2.4 Objective 4: Attitude Estimation	5
1.3 Contributions	5
1.3.1 Separation of Bias and Attitude Estimation	6
1.3.2 Manual Re-initialization of Attitude	6
1.3.3 Demonstration of Approach via Simulation and Field Experiments	7
2 Literature Review	8
2.1 History of Optical Attitude Estimation on Crewed Spacecraft	8
2.1.1 Gemini Program	8
2.1.2 Apollo Program	12
2.1.3 Skylab Space Station	13
2.1.4 Space Shuttle Program	15
2.1.5 Summary of Crew-Operated Optical Devices	15
2.2 The Wahba Problem	16
2.2.1 Derivation of the Wahba Problem as a Maximum Likelihood Estimate	17
2.2.2 Derivation of the Optimal Solution to Wahba’s Problem via SVD	19
2.3 Numerical Optimization Techniques	22
2.3.1 Linear Least Squares	22
2.3.2 Unconstrained Optimization	24
2.3.3 Newton’s Method in Optimization	26
2.3.4 Quasi-Newton Method	27
2.3.5 Levenberg-Marquardt Algorithm	29
2.3.6 Simulated Annealing Algorithm	32
2.3.7 Grid Search Technique	33
2.3.8 Multiplicative Extended Kalman Filter	35
3 Research Approach and Methodology	38
3.1 MOAR Algorithm Overview	38
3.2 The J2000 Reference System	40
3.3 Star Catalog	41
3.3.1 Effect of Parallax on Apparent Star Directions	42

3.3.2	Effect of Stellar Aberration on Apparent Star Directions	43
3.4	Manual Scope Measurements	44
3.5	IMU Measurements	45
3.5.1	Angle Random Walk	47
3.5.2	IMU Bias Instability	48
3.5.3	Allan Variance	48
3.6	Gyro Bias Estimation	49
3.6.1	Cost Function Definition	50
3.6.2	Unconstrained Optimization Implementation	51
3.6.3	Levenberg-Marquardt Implementation	54
3.6.4	Simulated Annealing Implementation	56
3.6.5	Optimization Using a Grid Search Algorithm and BFGS	57
3.6.6	Multiplicative EKF Implementation	59
3.6.7	Selecting an Optimization Routine for Use in the MOAR Algorithm	61
3.6.8	Effect of Bias Instability on Bias Estimates	64
3.6.9	Effect of Using Different Constellations for Star Sightings	66
3.6.10	Effect of Sampling Stars in Different Orders	68
3.6.11	Effect of Time Between Measurements on Solution Space	70
3.7	Attitude Estimation	70
3.8	Simulation of the MOAR Algorithm	72
3.8.1	Simulating Device Motion for a Sequenced Number of Stars	73
3.8.2	Determining Initial Attitude	74
3.8.3	Pointing Error Modeling	74
4	Experimental Approach and Design	76
4.1	Experimental Testing Apparatus	76
4.1.1	Data Acquisition	78
4.1.2	Inertial Measurement Unit	80
4.2	IMU Calibration	80
4.3	Uncertainty Budget	83
4.4	Experimental Procedure	85
5	Results	87
5.1	Monte Carlo Simulation	87
5.2	Results from MOAR Prototype Testing	89
5.2.1	Test Case 1: Big Dipper	89
5.2.2	Test Case 2: Orion	91
5.3	Summary of Results	94
6	Conclusions and Recommendations	96
6.1	Conclusions	96
6.2	Recommendations for Future Work	97
	Appendices	106
	Appendix A- IAU Constellation Charts	107
	Appendix B- MOAR Data Collection Notes	109
	Appendix C- MOAR Prototype Code	111

List of Figures

1.1	A concept drawing of the Orion capsule in space, courtesy of NASA	3
2.1	NASA space sextant (left) and USAF stadimeter (right) used for Guidance and Navigation onboard Apollo, courtesy of NASA.	9
2.2	The star occultation photometer onboard Gemini X, courtesy of NASA.	10
2.3	Star occultation photometer output for an observation of Vega. At point A, the apparent intensity of Vega began to diminish as the LOS began to pass through the Earth's atmosphere. It then disappears into the background between points B and C. Figure courtesy of NASA.	10
2.4	Illustration of astronaut performing star sightings using the sextant onboard the Gemini spacecraft, reprinted with permission of the Institute of Navigation.	11
2.5	Standard deviation, in arcsec, of Gemini sextant sighting measurements versus telescope magnification from, reprinted with permission of the American Institute of Aeronautics and Astronautics, Inc.	11
2.6	Schematic of the space sextant (left) and scanning telescope (right) used for Guidance and Navigation onboard Apollo, courtesy of NASA.	13
2.7	Graphical representation of how the stadimeter was used to measure the apparent curvature of a celestial body for ranging (left), and geometric representation of a stadimeter angle measurement (right), courtesy of USAF.	14
2.8	A plot used to determine a spacecraft's altitude above the observed celestial body, courtesy of USAF.	14
2.9	Timeline of crew-operated optical devices for navigation and their concurrent NASA human spaceflight programs	16
2.10	Process flowchart of the Simulated Annealing algorithm	33
2.11	Surface plot of the x-axis solution space associated with the Big Dipper and 10 seconds between measurements. A grid is superimposed over the solution space to simulate candidate bias guess subspaces.	34
3.1	Process flowchart of the MOAR algorithm	39
3.2	J2000 Equatorial Coordinate System, inspired by diagram from Navipedia	41
3.3	Star LOS vector defined by right ascension α and declination δ	42
3.4	Illustration of the effect of parallax on the closest star to the Solar System, Alpha Centauri	42
3.5	Example of mapping the rotation between three star sightings	45
3.6	Solution space of the bias estimation cost function with a step size between measurements of 10 seconds.	51
3.7	Process Flowchart for the Bias Estimation Routine	52
3.8	BFGS bias estimates using <i>fminunc</i> function in Matlab, with a red diamond denoting the true bias (left) and error histograms of the BFGS bias estimates (right)	53
3.9	LM bias estimates using <i>lsqnonlin</i> function in Matlab, with a red diamond denoting the true bias (left) and error histogram of the LM bias estimates (right)	55
3.10	Simulated Annealing bias estimates using <i>simulannealbnd</i> function in Matlab, with a red diamond denoting the true bias (left) and error histogram of the SA bias estimates (right)	57

3.11	BFGS-grid bias estimates (left) and error histogram of bias estimates found using the BFGS-grid algorithm (right)	58
3.12	Bias (left) and attitude (right) error plots from the MEKF Monte Carlo simulation with good <i>a priori</i> knowledge of bias and attitude. Bias and attitude estimates over time for all 200 Monte Carlo results are in gray with 3σ covariance bounds in black.	59
3.13	Bias (left) and attitude (right) error plots from the MEKF Monte Carlo simulation with very poor <i>a priori</i> knowledge of bias and attitude. Bias and attitude estimates over time for all 200 Monte Carlo results are in gray with 3σ covariance bounds in black.	60
3.14	The gyro bias estimates for 200 Monte Carlo runs (blue stars) are overlayed on contours of cost function value to show the performance of the various optimization methods under consideration. The true bias is $\mathbf{b} = [2.86 \ 2.86 \ 2.86]^T$ deg/s. Only the BFGS-Grid method is able to handle the highly nonlinear and multimodal cost function in a robust manner.	62
3.15	Final cost function values for all four optimization techniques versus final iteration number. It is apparent that the BFGS-grid technique had the lowest average cost function values with the number of iterations required consistent with that of SA. Because of this it was chosen for use in the MOAR algorithm.	63
3.16	Simulated performance of MEMS, tactical, and navigational grade IMUs in regard to bias instability. The true bias is $\mathbf{b} = [2.86 \ 2.86 \ 2.86]^T$ deg/s.	64
3.17	Bias estimate error histograms for MEMS, tactical and navigation grade IMUs	65
3.18	Allan variance plot for the ADIS16405 IMU reproduced using a bias instability 1σ of 0.007 deg/s and a 1σ noise value of 0.125 deg/s	66
3.19	Star plots of all constellations used overlaid with the order of star sightings (sequence of the trace pattern).	67
3.20	Bias estimate error histograms for three well distributed star patterns: Big Dipper (top row), Cygnus (middle row), and Orion (bottom row).	68
3.21	Bias estimate error histograms for a nearly degenerate star pattern: Aries. Since the solution space is multi-modal, in the presence of noise there exist multiple solutions when using a nearly degenerate star pattern such as Aries.	68
3.22	Bias error histogram from using the Big Dipper, starting with the first star and tracing to the right (left) and starting with the last star and tracing to the left (right)	69
3.23	Bias error histogram from using the Big Dipper, starting with the middle star and making a right handed system, no noise (left) and noisy (right)	69
3.24	Contour plots showing how the solution space cost function values change as time between star sightings changes	71
3.25	Cross Product Between two Star Vectors	73
3.26	Exaggerated visualization of pointing error when targeting stars in the Big Dipper. Red lines and circles represent true LOS measurements, and green lines and circles represent pointing error.	75
4.1	Engineering drawing of the MOAR prototype (left) and constructed MOAR prototype (right)	77
4.2	Illuminated reticle of the scope used on the MOAR prototype	77
4.3	Engineering drawing of Raspberry pi camera with light baffle and case (left) and rapid prototyped build (right)	78
4.4	Edmund Optics camera used on the MOAR prototype	78
4.5	Raspberry Pi Model B used on the MOAR prototype	79
4.6	Engineering drawing of the printed circuit board used on the MOAR prototype	79
4.7	Axial orientation of the ADIS16405 IMU and the MOAR sensor frame	80
4.8	MOAR prototype electronics mounted inside the ASEL's patent-pending calibration rig on a single axis rate table, aligned with the IMU <i>z</i> -axis (same as the MOAR <i>y</i> -axis)	81
4.9	Rate table IMU calibration results for the x gyro output, in the local IMU reference frame.	82
4.10	Rate table IMU calibration results for the y gyro output, in the local IMU reference frame.	82
4.11	Rate table IMU calibration results for the z gyro output, in the local IMU reference frame.	83
4.12	Experimental setup of the MOAR prototype on March 2, 2016	86

5.1	Attitude error histogram of a Monte Carlo simulation of the MOAR algorithm	88
5.2	Visualization of all ten star tracings (gray lines) of the Big Dipper plotted in the inertial frame using raw test data (top), data with steady-state bias removed (middle), and bias-corrected data (bottom). The black dots represent true LOS directions from the star catalog.	90
5.3	Visualization of all ten star tracings (gray lines) of the Big Dipper plotted in the inertial frame using raw test data (top), data with steady-state bias removed (middle), and bias-corrected data (bottom). The black dots represent true LOS directions from the star catalog.	93
6.1	IAU constellation charts for Aries (top left), Ursa Major (top right), Orion (bottom left), and Cygnus (bottom right)	108

List of Tables

2.1	Summary of the capabilities of various onboard manual navigation devices flown on NASA missions	16
3.1	Termination conditions for <i>fminunc</i> , returned as integer	53
3.2	Termination conditions for <i>lsqnonlin</i> , returned as integer	56
3.3	Maximum, minimum, mean and standard deviation (σ) values for the bias estimates of all 4 optimization routines. The true bias for each axis was 2.8648 <i>deg/s</i>	63
4.1	Worst case uncertainty budget for star sighting directions for the MOAR prototype	84
5.1	Summary of error values for the MOAR algorithm Monte Carlo simulation	87
5.2	Maximum, minimum, mean, and standard deviation (σ) values for the bias estimates of all four optimization routines	88
5.3	Cost function values for raw data, raw data with steady-state (SS) bias removed, and MOAR bias corrected data for the Big Dipper	91
5.4	Steady-state (SS) bias values derived from IMU output at the beginning of each Big Dipper star sighting compared against optimal estimate of the bias from the MOAR algorithm, in <i>deg/s</i> , compared to the final cost function value	91
5.5	Cost function values for raw data, raw data with steady-state (SS) bias removed, and MOAR bias corrected data for Orion	92
5.6	Steady-state (SS) bias values derived from IMU output at the beginning of each Orion star sighting compared against optimal estimate of the bias from the MOAR algorithm, in <i>deg/s</i>	92
5.7	Average cost function values of the Big Dipper and Orion tests	94
6.1	Start, stop and elapsed time data from MOAR prototype experiment 1: Big Dipper, March 2, 2016. Run numbers with asterisks denote bad measurement data due to user error.	109
6.2	Start, stop and elapsed time data from MOAR prototype experiment 2: Orion, March 9, 2016. Run numbers with asterisks denote bad measurement data due to user error. . . .	109
6.3	Start, stop and elapsed time data from MOAR prototype experiment 2: Orion, March 11, 2016. Run numbers with asterisks denote bad measurement data due to user error. . . .	110

Nomenclature

AC	Alternating Current
AIAA	American Institute of Aeronautics and Astronautics
ASEL	Applied Space Exploration Lab
BFGS	Broydon-Fletcher-Goldfarb-Shanno
BSC	Bright Star Catalog
CME	Coronal mass ejection
COAS	Crew Optical Alignment Sight
CPU	Central processing unit
DFP	Davidon-Fletcher-Powell
DoF	Degrees of freedom
FOV	Field of view
GNC	Guidance, Navigation and Control
HF	High frequency
HOT	Higher order terms
HWS	Holographic weapon sight
IMU	Inertial Measurement Unit
ION	Institute of Navigation
ISS	International Space Station
LEO	Low Earth Orbit
LLS	Linear Least Squares
LM	Levenberg-Marquardt
LOS	Line-of-sight
MEKF	Multiplicative Extended Kalman Filter
MEMS	Microelectromechanical systems
MOA	Minutes of Arc
MOAR	Manual Optical Attitude Re-initialization
NASA	National Aeronautics and Space Administration
QUEST	Quaternion Estimator
SA	Simulated Annealing
SVD	Singular Value Decomposition
USAF	United States Air Force
VHF	Very high frequency
A	Matrix used in The Matrix Equation
b	Vector used in the Matrix Equation
B	Attitude profile matrix
C	The set of complex numbers
C^n	n-dimensional complex space
dJ	First differential of the cost function J
d^2J	Second differential of the cost function J
$d\mathbf{x}$	First differential of the solution of LLS
$diag()$	diagonal matrix of ()
$eigenvect()$	eigenvectors of ()

f	Objective function
f'	First derivative of f
f''	Second derivative of f
$\mathbf{H}$	Hessian matrix
I	Inertial frame
$\mathbf{I}$	Identity matrix
J	Cost function
$\mathbf{K}$	Kalman gain
L	number that Newton's method converges to
$\mathbf{M}$	Diagonal matrix used to solve for optimal attitude matrix
N	Neighborhood
N	Order of convergence in Newton's method
ppm	parts per million
$\mathbf{r}$	Vector of residuals in the LLS problem
$\mathbf{R}$	Measurement covariance matrix
s_i	i^{th} eigenvalue of the Attitude profile matrix $\mathbf{B}$
$\mathbf{s}_k$	Search direction at the k^{th} iteration in the BFGS algorithm
S	Set of convex combinations of pairs of points
S_i	Sensor frame of the j^{th} measurement
S_j	Sensor frame of the j^{th} measurement
$\mathbf{S}$	Diagonal matrix found from SVD
$\mathbf{T}$	3×3 Rotation matrix
$\mathbf{U}$	Matrix found from SVD
$\mathbf{U}_{k-1}$	Rank-one update matrix in the BFGS algorithm
$\mathbf{V}$	Matrix found from SVD
$\mathbf{V}_{k-1}$	Rank-one update matrix in the BFGS algorithm
$\mathbf{u}$	unit vector
$\mathbf{u}_{i_I}$	i^{th} Line-of-sight vector in the inertial frame
$\mathbf{u}_{i_{S_j}}$	i^{th} Line-of-sight vector in the sensor frame
$\mathbf{W}$	Matrix used in derivation of the Wahba problem
$\mathbf{x}$	Solution to The Matrix Equation
x	Point used in unconstrained optimization background
X	Domain of feasible points in unconstrained optimization
y	Point used in unconstrained optimization background
$\mathbf{y}$	Model of the curve to be fit in the LM algorithm
$\mathbf{y}_k$	Vector of difference in gradient at the k^{th} iteration in the BFGS algorithm
α	Right Ascension
α_k	Step size at the k^{th} iteration in the BFGS algorithm
δ	Declination
$\delta\phi$	Rotation error vector
$\delta\mathbf{T}$	Rotation error matrix
Δt	Change in time
Δx	Incrementally small change in x
ϵ	Noise on LOS measurement in Wahba derivation
μ	rate of convergence in Newton's method
E	Expected value operator
λ	Levenberg-Marquardt trust-region scaling factor
λ_0	Scalar value used in SVD cost function
μ	rate of convergence in Newton's method
∇J	Gradient of the cost function
$\nabla^2 J$	Hessian of the cost function
ν	Gyro noise
ϕ	Rotation angle

σ	Standard deviation
Σ	sum
ω	Gyro data
$\ \cdot \ $	Matrix 2-norm operator
$cond(\cdot)$	Condition number of some matrix
$tr[\cdot]$	Trace operator
$\otimes$	Quaternion multiplication
$\mathbb{R}$	All real numbers
$\mathbb{R}^n$	n^{th} dimensional real space
$\forall$	For all
$\in$	Is an element of
$\subseteq$	Is a subset of
$\tilde{\bullet}$	Noisy measurement
$\hat{\bullet}$	Optimal Estimate
$\bullet_{i_I}$	i^{th} measurement in the inertial frame
$\bullet_{i_{S_j}}$	i^{th} measurement in the sensor frame
$\bullet_+$	Proper orthogonal matrix
$\bullet_\phi$	associated with rotation angle vector in Wahba derivation
$\bullet^T$	Transpose
$\bullet^*$	globally optimal solution

Chapter 1

Introduction

With the launch of crewed deep space missions in the near future, accurate and redundant attitude estimation systems are imperative for mission success. Current optical navigation technology makes use of line of sight (LOS) measurements to stars and other celestial bodies for position and/or attitude determination. The Gemini and Apollo programs laid the groundwork for development of a space sextant that was capable of calculating attitude estimates using star sightings. This provided astronauts with a contingency device that could be used in the event of a communications failure with ground stations on Earth. Other devices were used that measured star occultation from the Earth's atmosphere and the amount of horizon of a planet in view as a way of determining orbital parameters. Most recently, the Space Shuttle program made use of a device called the Crew Optical Attitude Sight (COAS) that could use star sightings as a way to correct IMU misalignment error down to acceptable levels in the event that it grew too large for star trackers to make accurate measurements. None of these devices, however, would be effective in re-initializing attitude in the event of a flight computer or star tracker malfunction where the spacecraft is considered to be lost in space. This is because the space sextant from Gemini and Apollo was intended to record attitude corrections, and the COAS does not take into account the bias on the gyro signal from the IMU. If the COAS were used to re-initialize attitude, the estimate it computed would be corrupted by this bias.

This work considers a device that makes use of star LOS measurements and gyro data from a low cost IMU to manually re-initialize a spacecraft's attitude. The Manual Optical Attitude Re-initialization (MOAR) device is designed to record the rotation between star sightings for a group of stars, remove bias from the gyro signal, then estimate attitude from the bias-corrected gyro data and LOS measurements. This is made possible by using gyro data between star sightings to compute interior star angles, which can then be compared against those computed from star catalog data to determine gyro bias. Since there are a myriad of techniques, several were selected and compared in their effectiveness of determining bias.

The bias can then be removed from the gyro data and the attitude estimate can be solved for.

A computer model of the device was constructed to determine the effectiveness of the concepts behind it. Star catalog data were used to generate simulated LOS, and the Farrenkopf model was used to simulate gyro measurements. Once the measurements were obtained, an analysis was performed that assessed four different optimization techniques' effectiveness in estimating gyro bias. From these results, a hybrid Grid Search/Quasi-Newton technique was chosen over the others and another simulation was performed that estimated attitude by solving the well-known Wahba Problem. A Monte Carlo simulation was then performed on the entire MOAR algorithm to determine its overall effectiveness in determining spacecraft attitude from noisy and biased gyro data and star LOS measurements.

A prototype of the MOAR device was constructed to experimentally validate the proposed MOAR algorithm. It consisted of a rifle scope, a computer vision camera, a MEMS grade IMU, and a Raspberry Pi. To use the device, first the user determined a group of stars to be targeted. The first star was then targeted with the scope, and a button was pressed to 'mark' the LOS measurement. The user then targeted the next star in the group, pressing the button to mark the star LOS measurement and the rotation between the two stars. This process was repeated for all stars in the group. The time history of recorded gyro data were then processed in the MOAR algorithm to determine a bias estimate. In the experiment, two groups of stars were chosen — the Big Dipper (of Ursa Major) and Orion. Both experiments were conducted with no prior training, and the results from each show that the MOAR algorithm is very effective at estimating bias and attitude. Its ease of use can be improved further with future work, allowing for simple implementation and use on a space vehicle.

1.1 Motivation

NASA and other agencies are showing increasing interest in sending astronauts past Low Earth Orbit (LEO), such as NASA's Orion project which can be seen in Figure 1.1. It will send astronauts farther than mankind has ever been before, and with that comes a lot of risk. For crew vehicles in LEO — such as missions to the International Space Station (ISS) — position and velocity states are computed on Earth and then sent to the spacecraft through satellite-ground station telecommunication link. As spacecraft travel farther from Earth, communication time required to receive updates increases. Because of this, deep space missions would be vulnerable to larger state estimation error between updates. Other issues that could arise with interplanetary missions include flight computer malfunctions and star tracker failures or obstructions. In the event of a star tracker or flight computer failure, a spacecraft must have a redundant method of determining its attitude. Previously, the Gemini and Apollo missions made use

of a stadimeter, space sextant, and scanning telescope that could compute updates that were supplied to the flight computer. These missions, however, were only concerned with cislunar navigation, and they were used in a secondary role for Command Module navigation.

Figure 1.1: A concept drawing of the Orion capsule in space [1], courtesy of NASA

While star tracker and flight computer failures are infrequent, they are still a real threat to space missions. The threat is even greater for deep space missions, where communication times are significantly longer with Earth. Space weather and solar events, such as solar flares and coronal mass ejections (CME), can and have caused anomalies onboard spacecraft. In October of 2003, a solar flare caused the satellite RHESSI's central processing unit (CPU) to reset on 3 separate instances. The CHIPS satellite's computer also went offline, and contact with Earth was lost for 18 hours. High and Very High Frequency (HF and VHF) communications can also be destroyed or severely downgraded during a solar event. [10] Radiation from solar events can also add noise to star trackers, corrupting and sometimes overwhelming the image. [11] Other space weather hazards include high energy electrons, which can penetrate through spacecraft walls and electronic boxes and embed themselves into dielectric materials, damaging not just GNC hardware but potentially other subsystems as well. [12] With these concerns in mind, it is imperative that astronauts onboard a spacecraft have a way to re-initialize attitude if the star trackers become disabled.

Work has been conducted previously on a device [13] that can be used onboard manned space missions to determine spacecraft attitude. It makes use of a military grade holographic weapons sight (HWS) that implements the desired reticle inspired by the Crew Optical Alignment Sight (COAS) reticle from previous Space Shuttle missions. Similarly to the COAS, it works by rotating the spacecraft to target stars, and marking the gimbal angles of the IMUs when the desired stars passes through the reticle. However, this process only aims to realign the IMUs in the event that the error exceeds a certain threshold. The onboard star trackers then take over after reducing the IMU error and finish the realigning process. Depending on the quality of the IMU, this device may not be able to accurately

re-initialize the spacecraft’s attitude if the gyro bias is significant and uncorrected. The MOAR device first removes bias from the gyro signal, and then calculates an attitude estimate without any *a priori* attitude estimate. The ability to separate bias from the gyro signal is critical to accurately solving the lost in space problem.

It will be beneficial to the space industry to have a technology to manually update and re-initialize attitude estimates past LEO. Such a device will provide additional safeguards against communication equipment or flight computer failure on-board the spacecraft. Orion will be the first vehicle that could potentially take advantage of this new technology. Using this as motivation, the MOAR device was designed to make use of an IMU and spotting scope to record measurements between star sightings, which can then be compared to corresponding stars from a catalog to determine attitude. This work explains the process behind the device and showcases its effectiveness as an adequate method to re-initialize a spacecraft’s attitude.

1.2 Research Objectives

With NASA and other agencies showing an increased interest in manned spaceflight past LEO, a way to manually calculate spacecraft attitude is being sought in the event of a flight computer malfunction. The MOAR device is designed to make use of an IMU and spotting scope to target stars and record the rotation between them. By comparing this data to a star catalog, the attitude can be calculated by solving the Wahba problem. However, the measurements from the IMU are inherently biased, and without correcting for this the attitude estimate will not be accurate. In order to realize a functioning device, the following objectives have been identified.

1.2.1 Objective 1: Numerical Simulation of the System

The MOAR algorithm is implemented in MATLAB[®] to simulate its functionality. Data on star positions is extracted from a star catalog for use in the simulation. A sequence of star sightings for a particular constellation is then simulated. The attitude and gyro data truth for each star is computed; noise and bias are added later to simulate measurement data. Gyro bias is estimated through the use of an optimization routine and removed to arrive at a measurement gyro estimate. This is then be used to calculate LOS vectors which simulate scope measurements, and both sets of measurement data are used to compute attitude. The methods are then verified with a Monte Carlo simulation.

1.2.2 Objective 2: Removal of Gyro Bias

Before spacecraft attitude can be determined, bias on the gyro measurements must be estimated and removed. In this work, several optimization methods are analyzed on their effectiveness in solving gyro bias, specifically a Quasi-Newton algorithm, the Levenberg-Marquardt (LM) method, and Simulated Annealing (SA). The gyro bias estimation is done completely independent of attitude estimation, which is a novel approach to this type of problem. A commonly used estimation technique in the Multiplicative Extended Kalman filter (MEKF) is also analyzed for its effectiveness in determining bias in sequential application. After analyzing the estimation methods, the best one is selected and implemented in removal of gyro bias.

1.2.3 Objective 3: Determine Importance of Accuracy vs. Time of Bias Estimate

When selecting an estimation technique, an important criterion to consider is the value of speed versus accuracy. Local optimization techniques converge to a solution quickly, but are susceptible to the initial value supplied to it. Global optimization techniques offer a way to determine a more accurate estimate but with a much longer solving time. The importance of speed versus accuracy in the MOAR algorithm is determined by comparing the quality of the estimates found using each estimation technique.

1.2.4 Objective 4: Attitude Estimation

Once the bias is removed from the gyro measurements, it is possible to accurately estimate spacecraft attitude. LOS vectors corresponding to the targeted stars may then be modeled from the bias-corrected gyro data and compared to the LOS vectors from the star catalog. There are many different methods for solving for attitude, such as using a Kalman filter [14], or solving the Wahba Problem [15] to name a few. Because of the nonlinear nature of the problem and the need for batch estimation, not all methods are applicable. In this dissertation, nonlinear attitude estimation is performed by solving the Wahba Problem.

1.3 Contributions

This work provides a detailed discussion on using an IMU and spotting scope to manually determine a spacecraft's attitude. Usually attitude determination is an autonomous function onboard the spacecraft, but this dissertation introduces a method for the crew to manually compute the data necessary to reset

the flight computer. The following sections cover in more detail each of the major contributions to the field that are presented in this dissertation.

1.3.1 Separation of Bias and Attitude Estimation

There are several instances in literature where bias has been estimated through the use of a Kalman filter, and it is typically done concurrently with attitude estimation. [16, 17] Nonlinear observers have also been used to determine gyro bias. [18] These methods assume to have some *a priori* knowledge of bias and/or attitude, which is critical in their implementation. This work attempts to solve the lost-in-space problem, which has no *a priori* knowledge of bias or attitude, and as such a Kalman filter may not be used.

To address this long-standing problem, a new technique was derived to estimate gyro bias without ever needing to actually know the attitude. The key is to solve for gyro bias using only interior star angles, a representation that remains the same regardless of the frame in which the actual star directions are expressed. For the MOAR device, these interior star angles were calculated by integrating noisy gyro data between star sightings with a scope. These measured interior star angles were then stored in a vector. In order to avoid errors from ‘flattening’, the angles for every non-repeating combination of stars were determined. The true angles between these same combinations of stars were also determined from the star catalog and stored in another vector. An optimization routine was then designed to minimize the difference between the two vectors by varying the bias at each iteration. The novel use of interior star angles makes the bias estimation problem completely separable from the attitude estimation problem. This makes a previously intractable problem easy to solve and represents a significant advancement to the state-of-the-art.

1.3.2 Manual Re-initialization of Attitude

Star trackers compute spacecraft attitude autonomously by measuring observed star directions in the sensor’s frame and comparing them to corresponding data from a star catalog. For a star tracker, which is essentially a camera, all these observations are concurrent since they are extracted from a single image. Conversely, the MOAR device permits a user to manually take sequential star measurements with a scope — all of which are mapped to a common frame using a gyro. The gyro bias removal from the first contribution enables the feasibility of this second contribution.

1.3.3 Demonstration of Approach via Simulation and Field Experiments

The research methodology laid out in this dissertation is proven through both simulated and empirical results. Once the MOAR algorithm was defined and implemented in MATLAB[®], it was simulated using 4 different constellations. The results from these Monte Carlo simulations showed that for a well distributed star pattern, the MOAR algorithm is very effective at estimating bias and attitude. It was also evident from another simulation that the order in which stars are sampled is unimportant, as long as they are consistent between being traced in practice and defined in the algorithm.

An experiment was then conducted using a prototype MOAR device, where 2 constellations — the Big Dipper and Orion — were used for data collection, and the data were post-processed in MATLAB[®]. The results from the experiment showed that the MOAR algorithm is very effective at estimating bias and attitude in practice. A significant takeaway from this was that no training is required to accurately operate the MOAR device. Its simple construction and implementation provides accurate estimates of bias and attitude that can be supplied to a spacecraft’s flight computer in the event of a navigation system/component casualty.

Chapter 2

Literature Review

A literature review was performed on the concepts used throughout this work to determine existing technologies. A review of attitude estimation methods used on previous NASA missions is presented, as well as numerical estimation techniques used to find an optimal bias estimate.

2.1 History of Optical Attitude Estimation on Crewed Spacecraft

NASA has used optical navigation techniques on many of their manned space flights. The Gemini missions were the first to test optical navigation. When Apollo flew, the previously researched technology was explored in further detail and tested in a cislunar navigation application, where it was very successful in achieving accurate attitude estimates derived from star sightings. The space shuttle program also made use of optical navigation, and an updated version of the Shuttle's technology has been analyzed for use on the Orion mission. The following sections discuss each program's progress toward the use of optical navigation in space.

2.1.1 Gemini Program

The Gemini program was NASA's second U.S. human spaceflight program, after the Mercury project which aimed at putting an astronaut into orbit around the Earth. Gemini used ground-based tracking as a primary means of navigation to ease the workload of the crew members and ensure high accuracy. As a contingency, optical navigation methods were proposed to provide an additional means of acquiring position and attitude estimates, yielding higher confidence regarding performance of the navigation

system. As a result, several optical navigation experiments were conducted by the US Air Force and NASA to determine the effectiveness of simple instruments in space.

The first experiment, Experiment D-009, examined the use of optical instruments in a space environment to which the results could be applied to solve orbital parameters [19]. This experiment was flown on Gemini IV and VII, and it called for the testing of a space sextant and stadimeter, seen in Figure 2.1. The space sextant experiment was successful, and it helped to lay the groundwork for further space sextant research, while the stadimeter was never actually flown during the Gemini program.

Figure 2.1: NASA space sextant (left) and USAF stadimeter (right) used for Guidance and Navigation onboard Apollo [2], courtesy of NASA.

The next experiment to be tested was Experiment D-005, flown on the Gemini VII and X missions, was intended to demonstrate the ability of astronauts to use star occultation measurements for space navigation [2]. To determine the occultation time of a star, a star occultation photometer [20], seen in Figure 2.2, determined the extent of which the LOS to a selected star penetrated the Earth's atmosphere by creating an attenuation curve of the star relative to an unattenuated intensity. The photometer onboard the Gemini VII mission had failed due to loose particles in the photomultiplier tube, and as a result no data were collected. After being refurbished and going through strict quality controls, the photometer functioned as designed on the Gemini X mission, the results of which can be seen in Figure 2.3. A problem was encountered where astronauts observed stars disappearing momentarily as they passed through the green-glow layer. Sighting was discontinued until the star reappeared, but excessive attitude-control gas expenditure caused from this procedure resulted in the experiment being canceled until docking was completed.

The third experiment, Experiment T-002, was conducted onboard an orbiting Gemini XII spacecraft,

Figure 2.2: The star occultation photometer onboard Gemini X [2], courtesy of NASA.

Figure 2.3: Star occultation photometer output for an observation of Vega. At point A, the apparent intensity of Vega began to diminish as the LOS began to pass through the Earth's atmosphere. It then disappears into the background between points B and C. Figure from [2], courtesy of NASA.

and was intended to determine the effectiveness of using a hand-held sextant for mid-course navigation using star-moon sightings [3, 4]. An illustration of the Gemini star-moon sightings tasks can be seen in Figure 2.4. The in-flight star sightings were performed by Buzz Aldrin; however, since the moon was not available as a target on the Gemini XII mission, the sighting data were obtained using star-star pairs. Results indicated that it was possible to obtain accurate star sightings (with a standard deviation of 10 *arcsec*) with and without the spacesuit helmet on. Another interesting point to note from the Gemini project is that as sextant magnification increased from 2-20X, there was a decrease in standard deviation of LOS measurements [21]. At 20X, the standard deviation of measurements was 5 *arcsec*, well below

the goal of 10 *arcsec* for mission goals.

Figure 2.4: Illustration of astronaut performing star sightings using the sextant onboard the Gemini spacecraft [3], reprinted with permission of the Institute of Navigation.

Figure 2.5: Standard deviation, in *arcsec*, of Gemini sextant sighting measurements versus telescope magnification from [4], reprinted with permission of the American Institute of Aeronautics and Astronautics, Inc.

Also worth noting is an experiment that was performed in 1967 at the Ames Midcourse Guidance and Navigation Simulator that aimed to study the usefulness of a hand-held space sextant used in rendezvous operations [22]. Measurements were taken between a flashing light on the desired rendezvous vehicle and a star. The flashing light target had variable flash characteristics, and various spacecraft rates were simulated to encompass a wide array of situations. The results of the experiment showed that with an increase in flash period by 5 to 20%, the measurement performance was significantly improved. Also, a training technique where the measurement error feedback was presented to the sextant operator showed measurement performance improvement. Overall, the experiment demonstrated that an experienced navigator using a hand-held sextant during rendezvous operations can expect to obtain measurements

with an error of less than 60 *arcsec*.

It should be noted that the type of orbital parameters that were solved for by these devices depended entirely on which celestial objects were used. For instance, a star-star pair was used to find an attitude estimate, while a star and another celestial body such as the Earth or Moon formed a pair that was used to estimate position relative to that particular celestial body. As a result, the space sextant could be used to determine either an attitude or position estimate based off of the type of measurement taken, while the star occultation photometer and stadimeter could only be used for determining the position of a spacecraft. Even though this work focuses solely on attitude estimation, both types of space navigation devices were covered because the process of collecting measurements using either of the pairs was ergonomically the same.

2.1.2 Apollo Program

The Apollo Command Module on-board Guidance Navigation and Control (GNC) system was envisioned as a self-contained capability to navigate a spacecraft for future deep space missions. Some of the reasons cited were signal transmission times between the spacecraft and a ground station, and potentially having many missions being conducted simultaneously would saturate expensive ground stations [23]. As the Apollo program progressed, it was determined that self-sufficient coding of the GNC system would be too complex and Mission Control in Houston would be able to perform the same functions while relieving the workload of astronauts on-board. The technology was still proven to be effective in cislunar navigation, although it required a lot of work on the part of the crew.

Apollo mid-course navigation was conducted through the use of a specially designed space sextant that measured the angle between any star and an Earth or Moon landmark. Astronauts manually operated the device by superimposing the targeted star on the landmark LOS. This device was capable of making accurate measurements on the order of 10 *arcsec*, when a 28X telescope was used [24]. Research by Duke and Charles also showed that accuracy increased as the telescope power was increased to 40X, with all errors below 7 *arcsec*.

The Optical Subsystem was equipped with a 28X sextant, which had a 1.8° field of view (FOV) instrument that required two LOS. One LOS, called the landmark LOS, was fixed to the spacecraft and could be pointed by turning the vehicle in space with the attitude control system. The other LOS, the star LOS, had 2 degrees of freedom (DoF) (the shaft and trunnion axes) and could be adjusted by either the computer or a two-axis hand controller. The shaft axis was parallel to the landmark LOS, and the trunnion axis could be controlled by rotating the trunnion-axis mirror. When a measurement was ready

to be recorded, the astronaut could press a button to ‘mark’ the event, which recorded the trunnion angle and the mark time. The Optical Subsystem also consisted of a one DoF scanning telescope, whose wide FOV of 60° aided in star identification. Operational schematics for both components may be seen in Figure 2.6. Both the space sextant and scanning telescope are covered in more detail in [5], [25], and [26].

Figure 2.6: Schematic of the space sextant (left) and scanning telescope (right) used for Guidance and Navigation onboard Apollo [5], courtesy of NASA.

Post-flight analysis showed that the Apollo 8 on-board system successfully navigated in cislunar space and demonstrated the self-contained return to Earth navigation capability needed in the event of loss of ground assistance [26]. Over 200 measurements were taken by the navigator command module pilot Jim Lovell, demonstrating the performance of the system.

2.1.3 Skylab Space Station

Skylab was launched in 1973 and was the United States’ first space station. The most notable experiment conducted aboard Skylab that analyzed the use of optical navigation in space was Experiment

T-002 [27]. This experiment analyzed the performance of a space sextant and stadimeter, and it was the first time that the stadimeter, seen in Figure 2.7, was flown on a mission to space. By taking three LOS measurements of the Earth's horizon, the altitude of the spacecraft, or in this case Skylab, could be determined. This is represented by the equation using the subtended half-angle of the Earth f ,

$$\sin f = \frac{R}{r} \quad (2.1.1)$$

where R is the radius of Earth, and r is the altitude of the spacecraft plus the radius of Earth. By measuring this angle, the altitude of the spacecraft could be determined from a plot, which can be seen in Figure 2.8 [6].

Figure 2.7: Graphical representation of how the stadimeter was used to measure the apparent curvature of a celestial body for ranging (left), and geometric representation of a stadimeter angle measurement (right) [6], courtesy of USAF.

Figure 2.8: A plot used to determine a spacecraft's altitude above the observed celestial body [6], courtesy of USAF.

2.1.4 Space Shuttle Program

The Space Shuttle onboard navigation system relied on star trackers and IMUs. However, it flew with an additional tool for contingency situations that require manual attitude corrections called the Crew Optical Alignment Sight (COAS) [28]. This device was used if and when IMU alignment was in error of 1.4° or greater, at which point the star trackers are unable to acquire and track stars. The COAS realigned the IMUs to within 1.4° ; once this was achieved, the star tracker was then used to realign the IMUs more precisely. This capability was used primarily to check for proper attitude orientation during ascent and de-orbit thrusting periods, where it was mounted at the commander's station. It was also used to monitor vehicle alignment in rendezvous and docking operations; however, it needed to be removed and installed next to the flight deck overhead minus-Z window.

The COAS is an optical device consisting of a spotting scope with a 10° FOV that has an illuminated reticle projected onto it. The reticle is aimed toward infinity and requires a 115 volt AC power source for variable brightness functionality. After removing the COAS from the commander's station and installing it at the aft flight station, astronauts could use the device to make manual LOS star measurements. This was done first by rotating the spacecraft until the stars were in view. When the desired star passed through the reticle, a 'mark' button was pressed, and the gimbal angles of the onboard IMUs were recorded. When it had been determined that a good mark was achieved, the software was notified to accept the measurement. A good star mark requires that stars be separated between 60 and 120 degrees.

By knowing the COAS's orientation relative to the spacecraft, which is calibrated before launch, it is possible to determine a LOS vector from the spacecraft to the star in an inertial reference system. Two stars are required to determine the attitude of the spacecraft in inertial space, which can then be compared to the attitude estimate calculated from the IMUs. If the IMUs are in error, the COAS estimate may be used to realign the axes to within the 1.4° of error threshold. Further work has been done to update the COAS for use on the Orion capsule by changing the spotting scope to a commercially available holographic weapons sight (HWS) and modifying its reticle to match that of the legacy COAS [13].

2.1.5 Summary of Crew-Operated Optical Devices

Gemini was the first NASA human spaceflight program, it proposed three optical attitude and/or position estimation devices, the space sextant, stadimeter and star occultation photometer. Throughout the program the stadimeter was never tested, while the space sextant and star occultation tests showed promising experimental results. The Apollo program further tested the space sextant, and showed an

The Wahba problem had arisen when estimating the attitude of a satellite using a set of vector observations in the satellite-fixed frame, and a set corresponding to the same objects in a known reference frame. Wahba then states that the matrix $\mathbf{T}_S^I$ is a least-squares estimate of the rotation matrix that maps the observations from the known frame to the satellite-fixed frame. This rotation matrix $\mathbf{T}_S^I$ is the satellite attitude estimate. A full derivation of the Wahba problem can be found in the next section.

2.2.1 Derivation of the Wahba Problem as a Maximum Likelihood Estimate

Suppose there exists a set of unit vector measurements to stars in the sensor frame, $\tilde{\mathbf{u}}_{i_S}$, and there also exists a corresponding set of unit vectors from a star catalog in the inertial frame, $\mathbf{u}_{i_I}$. This gives the measurement model,

$$\begin{aligned}\mathbf{u}_{i_S} &= \mathbf{T}_S^I \mathbf{u}_{i_I} \\ \tilde{\mathbf{u}}_{i_S} &= \mathbf{T}_S^I \mathbf{u}_{i_I} + \epsilon \\ \tilde{\mathbf{u}}_{i_S} &= \mathbf{u}_{i_S} + \epsilon\end{aligned}\tag{2.2.2}$$

where ϵ is error on the LOS vectors and $E[\epsilon] = 0$. The equations above can be expressed in terms of an error rotation matrix $\delta\mathbf{T}$,

$$\tilde{\mathbf{u}}_{i_S} = \delta\mathbf{T}\mathbf{u}_{i_S}\tag{2.2.3}$$

A rotation matrix may be approximated as the matrix exponential of $[-\delta\phi \times]$, where $\delta\phi$ is the rotation expressed as an angle vector. Therefore,

$$\delta\mathbf{T} = \exp([-\delta\phi \times]) \approx \mathbf{I}_{3 \times 3} + [-\delta\phi \times] + \frac{1}{2}[-\delta\phi \times]^2 + H.O.T\tag{2.2.4}$$

Taking $\delta\mathbf{T}$ to first order,

$$\begin{aligned}\tilde{\mathbf{u}}_{i_S} &= (\mathbf{I}_{3 \times 3} + [-\delta\phi \times])\mathbf{u}_{i_S} \\ &= \mathbf{u}_{i_S} - [\delta\phi \times]\mathbf{u}_{i_S} \\ &= \mathbf{u}_{i_S} + [\mathbf{u}_{i_S} \times]\delta\phi\end{aligned}\tag{2.2.5}$$

Recalling that $\tilde{\mathbf{u}}_{i_S} = \mathbf{u}_{i_S} + \epsilon$,

$$\epsilon = [\mathbf{u}_{i_S} \times] \delta \phi \quad (2.2.6)$$

Next, a covariance matrix $\mathbf{R}$ can be defined as

$$\mathbf{R} = E[(\tilde{\mathbf{u}}_{i_S} - E[\tilde{\mathbf{u}}_{i_S}])(\tilde{\mathbf{u}}_{i_S} - E[\tilde{\mathbf{u}}_{i_S}])^T] \quad (2.2.7)$$

which is rank-2 because $\mathbf{u}_{i_S}$ is unconstrained; therefore, $\mathbf{R}$ is non-invertible. To simplify,

$$E[\tilde{\mathbf{u}}_{i_S}] = E[\mathbf{u}_{i_S} + \epsilon] \quad (2.2.8)$$

where ϵ represents zero-mean white noise with $E[\epsilon] = 0$ so,

$$E[\tilde{\mathbf{u}}_{i_S}] = \mathbf{u}_{i_S} \quad (2.2.9)$$

Using the definition of a measurement covariance matrix, $\mathbf{R} = E[\epsilon \epsilon^T]$,

$$\begin{aligned} \mathbf{R} &= E[[\mathbf{u}_{i_S} \times] \delta \phi \delta \phi^T [\mathbf{u}_{i_S} \times]^T] \\ &= [\mathbf{u}_{i_S} \times] E[\delta \phi \delta \phi^T] [\mathbf{u}_{i_S} \times]^T \end{aligned} \quad (2.2.10)$$

and

$$E[\delta \phi \delta \phi^T] = \sigma_\phi^2 \mathbf{I}_{3 \times 3} \quad (2.2.11)$$

Therefore,

$$\begin{aligned} \mathbf{R} &= \sigma_\phi^2 [\mathbf{u}_{i_S} \times] [\mathbf{u}_{i_S} \times]^T \\ &= -\sigma_\phi^2 [\mathbf{u}_{i_S} \times] [\mathbf{u}_{i_S} \times] \\ &= -\sigma_\phi^2 (\mathbf{u}_{i_S} \mathbf{u}_{i_S}^T - \mathbf{u}_{i_S}^T \mathbf{u}_{i_S} \mathbf{I}_{3 \times 3}) \\ \mathbf{R} &= \sigma_\phi^2 (\mathbf{I}_{3 \times 3} - \mathbf{u}_{i_S}^T \mathbf{u}_{i_S}) \end{aligned} \quad (2.2.12)$$

where $\mathbf{R}$ is sometimes called the QUEST covariance matrix. Now consider the following Maximum Likelihood Estimation (MLE) problem,

$$\text{Min } J(\mathbf{T}_S^I) = \frac{1}{2} \sum (\mathbf{u}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I})^T \mathbf{R}_i^{-1} (\mathbf{u}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I}) \quad (2.2.13)$$

$$\text{Min } J(\mathbf{T}_S^I) = \frac{1}{2} \sum \epsilon_i^T \mathbf{R}_i^{-1} \epsilon_i \quad (2.2.14)$$

$$\text{Min } J(\mathbf{T}_S^I) = \frac{1}{2} \sum \epsilon_i^T \left(\frac{1}{\sigma_\phi^2} (\mathbf{I}_{3 \times 3} - \mathbf{u}_{i_S} \mathbf{u}_{i_S}^T) \right) \epsilon_i \quad (2.2.15)$$

$$\text{Min } J(\mathbf{T}_S^I) = \frac{1}{2} \sum \frac{1}{\sigma_\phi^2} (\epsilon_i^T \epsilon_i - \epsilon_i^T \mathbf{u}_{i_S} \mathbf{u}_{i_S}^T \epsilon_i) \quad (2.2.16)$$

since $\mathbf{u}_{i_S}^T \epsilon = 0$,

$$\text{Min } J(\mathbf{T}_S^I) = \frac{1}{2} \sum \frac{1}{\sigma_\phi^2} (\epsilon_i^T \epsilon_i) \quad (2.2.17)$$

Recall that $\epsilon_i = \tilde{\mathbf{u}}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I}$,

$$\text{Min } J(\mathbf{T}_S^I) = \frac{1}{2} \sum \frac{1}{\sigma_\phi^2} (\tilde{\mathbf{u}}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I})^T (\tilde{\mathbf{u}}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I}) \quad (2.2.18)$$

This can be reduced to the final form of the Wahba Problem,

$$\text{Min } J(\mathbf{T}_S^I) = \frac{1}{2} \sum \frac{1}{\sigma_\phi^2} \|\tilde{\mathbf{u}}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I}\|^2 \quad (2.2.19)$$

2.2.2 Derivation of the Optimal Solution to Wahba's Problem via SVD

The following derivation shows that an optimal solution to Wahba's Problem may be solved by using SVD. For simplicity $\mathbf{T}_S^I$ will be replaced by $\mathbf{T}$. First, the final form of the Wahba Problem in Equation 2.2.19 can be expanded,

$$\text{Min } J(\mathbf{T}) = \frac{1}{2} \sum \frac{1}{\sigma_\phi^2} (\tilde{\mathbf{u}}_{i_S}^T \tilde{\mathbf{u}}_{i_S} - \tilde{\mathbf{u}}_{i_S}^T \mathbf{T} \mathbf{u}_{i_I} - \mathbf{u}_{i_I}^T \mathbf{T} \tilde{\mathbf{u}}_{i_S} + \mathbf{u}_{i_I}^T \mathbf{T}^T \mathbf{T} \mathbf{u}_{i_I}) \quad (2.2.20)$$

Recall that the inner product of a vector with itself is equal to one ($\mathbf{u}^T \mathbf{u} = 1$), and since $\mathbf{T}$ is orthonormal, $\mathbf{T}^T \mathbf{T} = \mathbf{T}^{-1} \mathbf{T} = \mathbf{I}$. Therefore,

$$\text{Min } J(\mathbf{T}) = \frac{1}{2} \sum \frac{1}{\sigma_\phi^2} (2 - 2\tilde{\mathbf{u}}_{i_S}^T \mathbf{T} \mathbf{u}_{i_I}) \quad (2.2.21)$$

$$\text{Min } J(\mathbf{T}) = \sum \frac{1}{\sigma_\phi^2} - \sum \frac{1}{\sigma_\phi^2} \tilde{\mathbf{u}}_{i_s}^T \mathbf{T} \mathbf{u}_{i_I} \quad (2.2.22)$$

Let $\sum \frac{1}{\sigma_\phi^2} = \lambda_0$, and since each term is a scalar the trace may be computed,

$$\text{Min } J(\mathbf{T}) = \lambda_0 - \sum \frac{1}{\sigma_\phi^2} \text{tr}[\tilde{\mathbf{u}}_{i_s}^T \mathbf{T} \mathbf{u}_{i_I}] \quad (2.2.23)$$

Using the cyclic invariance property of the trace operator $\text{tr}[\mathbf{ABC}] = \text{tr}[\mathbf{BCA}] = \text{tr}[\mathbf{CAB}]$,

$$\text{Min } J(\mathbf{T}) = \lambda_0 - \sum \frac{1}{\sigma_\phi^2} \text{tr}[\mathbf{T} \mathbf{u}_{i_I} \tilde{\mathbf{u}}_{i_s}^T] \quad (2.2.24)$$

Since the summation operator is a linear operator, it may be brought into the trace operator

$$\text{Min } J(\mathbf{T}) = \lambda_0 - \text{tr}\left[\mathbf{T} \left(\sum \frac{1}{\sigma_\phi^2} \mathbf{u}_{i_I} \tilde{\mathbf{u}}_{i_s}^T \right)\right] \quad (2.2.25)$$

Where $\mathbf{B}^T = \sum \frac{1}{\sigma_\phi^2} \mathbf{u}_{i_I} \tilde{\mathbf{u}}_{i_s}^T$ is the transpose of the attitude profile matrix from Equation 3.7.2

$$\text{Min } J(\mathbf{T}) = \lambda_0 - \text{tr}[\mathbf{T} \mathbf{B}^T] \quad (2.2.26)$$

The singular value decomposition of $\mathbf{B}$ can be found by,

$$\mathbf{B} = \mathbf{U} \mathbf{S} \mathbf{V}^T \quad (2.2.27)$$

where $\mathbf{U} = \text{eigenvect}(\mathbf{B} \mathbf{B}^T)$ and $\mathbf{V} = \text{eigenvect}(\mathbf{B}^T \mathbf{B})$ are orthogonal matrices, and $\mathbf{S} = \text{diag}(s_1, s_2, s_3)$ with $s_1 \geq s_2 \geq s_3 \geq 0$ [29]. Proper orthogonal matrices may then be defined by

$$\mathbf{U}_+ = \mathbf{U}[\text{diag}(1, 1, \det(\mathbf{U}))] \quad (2.2.28)$$

$$\mathbf{V}_+ = \mathbf{V}[\text{diag}(1, 1, \det(\mathbf{V}))] \quad (2.2.29)$$

and

$$\mathbf{W} = \mathbf{U}_+^T \mathbf{T} \mathbf{V}_+ = \cos \phi \mathbf{I} + (1 - \cos \phi) \mathbf{u} \mathbf{u}^T - \sin \phi [\mathbf{u} \times] \quad (2.2.30)$$

where $[\mathbf{u} \times]$ is the skew symmetric matrix corresponding to a unit vector $\mathbf{u}$. Equation 2.2.30 is an Euler axis/angle representation of $\mathbf{W}$ by a unit vector and rotation angle ϕ , and any proper orthogonal matrix

may be represented in this way. Then a diagonal matrix is defined,

$$\mathbf{S}' = \text{diag}(s_1, s_2, ds_3) \quad (2.2.31)$$

where

$$d = \det(\mathbf{U})\det(\mathbf{V}) = \pm 1 \quad (2.2.32)$$

Equation 2.2.27 may then be re-written,

$$\mathbf{B} = \mathbf{U}_+ \mathbf{S}' \mathbf{V}_+^T \quad (2.2.33)$$

and can then be substituted back into Equation 2.2.25 assuming $\lambda_0 = 1$,

$$\begin{aligned} \text{Min } J(\mathbf{T}) &= 1 - \text{tr}[\mathbf{S}'\mathbf{W}] \\ &= 1 - \text{tr}[\mathbf{S}'] + (1 - \cos \Phi)[s_2 + ds_3 + (s_1 - s_2)e_2^2 + (s_1 - ds_3)e_3^2] \end{aligned} \quad (2.2.34)$$

It is apparent that Equation 2.2.34 is minimized for $\Phi = 0$ which yields $\mathbf{W} = \mathbf{I}$,

$$\text{Min } J(\hat{\mathbf{T}}) = 1 - \text{tr}[\mathbf{S}'] = 1 - s_1 - s_2 - ds_3 \quad (2.2.35)$$

thus, the equation for the optimal attitude matrix, $\hat{\mathbf{T}}$ can be expressed by,

$$\hat{\mathbf{T}} = \mathbf{U}\mathbf{V}_+^T = \mathbf{U}\mathbf{M}\mathbf{V}^T \quad (2.2.36)$$

where

$$\mathbf{M} = \text{diag}\left\{ \begin{bmatrix} 1 & 1 & \det(\mathbf{U})\det(\mathbf{V}) \end{bmatrix} \right\} \quad (2.2.37)$$

Now that a method for attitude determination has been fully defined, the discussion will shift to finding the best method to remove gyro bias. In this work, numerical optimization techniques were analyzed as a way to estimate gyro bias independently from attitude.

2.3 Numerical Optimization Techniques

Many algorithms have been developed over the years that attempt to solve for the true optimal solution, but there are two factors in developing these algorithms that must be weighed against each other: speed and complexity. A standard unconstrained optimization routine aims to solve for the optimum as fast as possible, and it is a relatively simple process. Other algorithms, such as the Levenberg-Marquardt algorithm, are more complex, and therefore will take more time to compute the estimate. These two algorithms are examples of local optimization techniques, and they were evaluated for use in the MOAR algorithm. An entirely different class of techniques exist called global optimization techniques that attempt to solve for the global optimum by using the entire search space, which inherently requires a lot of time. Several global optimization techniques exist such as Genetic Algorithms and Simulated Annealing. In this work, Simulated Annealing was also evaluated for use in the MOAR algorithm. All three of these techniques will be discussed in depth below.

2.3.1 Linear Least Squares

Linear Least Squares (LLS) is an estimation technique that approximates the values of an over-determined system of the form $\mathbf{Ax} = \mathbf{b}$. Using this form, a solution for $\mathbf{x}$ that is as close to $\mathbf{b}$ as possible, which is represented by $\hat{\mathbf{x}}$. A solution for LLS may be derived by considering the optimization problem of minimizing the sum of the squares of residuals,

$$\mathbf{r} = \mathbf{b} - \mathbf{Ax} \quad (2.3.1)$$

Then the cost function is defined as,

$$\text{Min } J(\mathbf{x}) = \|\mathbf{b} - \mathbf{Ax}\|^2 = (\mathbf{b} - \mathbf{Ax})^T (\mathbf{b} - \mathbf{Ax}) \quad (2.3.2)$$

where $J(\mathbf{x})$ represents a cost function with the variable $\mathbf{x}$. It can then be expanded,

$$\text{Min } J(\mathbf{x}) = \mathbf{b}^T \mathbf{b} - \mathbf{b}^T \mathbf{Ax} - \mathbf{x}^T \mathbf{A}^T \mathbf{b} + \mathbf{x}^T \mathbf{A}^T \mathbf{Ax} \quad (2.3.3)$$

$$\text{Min } J(\mathbf{x}) = \mathbf{b}^T \mathbf{b} - 2\mathbf{b}^T \mathbf{Ax} + \mathbf{x}^T \mathbf{A}^T \mathbf{Ax} \quad (2.3.4)$$

The problem may then be solved by setting the first differential equal to zero,

$$dJ = (-2\mathbf{b}^T \mathbf{A} + 2\mathbf{x}^T \mathbf{A}^T \mathbf{A})d\mathbf{x} = 0 \quad (2.3.5)$$

since $d\mathbf{x}$ cannot equal zero,

$$-2\mathbf{b}^T \mathbf{A} + 2\mathbf{x}^T \mathbf{A}^T \mathbf{A} = 0 \quad (2.3.6)$$

$$\mathbf{x}^T \mathbf{A}^T \mathbf{A} = \mathbf{b}^T \mathbf{A} \quad (2.3.7)$$

Finally,

$$\mathbf{A}^T \mathbf{A} \hat{\mathbf{x}} = \mathbf{A}^T \mathbf{b} \quad (2.3.8)$$

Equation 2.3.8 is known as the Normal Equation, because the residuals are perpendicular (normal) to the columns of $\mathbf{A}$. In order for a minimum to exist, $d^2 J > 0$ must be true,

$$\begin{aligned} d^2 J &= 2d\mathbf{x}^T \mathbf{A}^T \mathbf{A} d\mathbf{x} > 0 \\ d\mathbf{x}^T \mathbf{A}^T \mathbf{A} d\mathbf{x} &> 0 \end{aligned} \quad (2.3.9)$$

Therefore, for there to be a minimum $\mathbf{A}^T \mathbf{A}$ must be positive definite,

$$\begin{aligned} \mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x} &> 0 \\ (\mathbf{A}\mathbf{x})^T \mathbf{A} \mathbf{x} &> 0 \\ \mathbf{b}^T \mathbf{b} &> 0 \end{aligned} \quad (2.3.10)$$

When solving linear systems of the form $\mathbf{A}\mathbf{x} = \mathbf{b}$, the ability to solve the problem with finite precision depends on the condition number of $\mathbf{A}$. Unfortunately when solving the normal equations, finite precision depends on the condition number of $\mathbf{A}^T \mathbf{A}$, and

$$\text{cond}(\mathbf{A}^T \mathbf{A}) = [\text{cond}(\mathbf{A})]^2 \quad (2.3.11)$$

Thus, by solving the normal equations, the condition number is squared with respect to the original problem. Along with this pitfall, LLS cannot be directly applied to the problem of determining spacecraft

attitude because the problem is inherently nonlinear. It may be applied if the nonlinear equations are first linearized about a point, however, this approach is computationally expensive. For nonlinear problems, an optimization routine may be performed instead.

2.3.2 Unconstrained Optimization

Unconstrained optimization problems are arguably the simplest type of optimization as they are defined by only an objective function [30]. The objective function (also referred to as the cost function) is then minimized, and the optimization problem may be represented by,

$$\text{Min } J(x) \tag{2.3.12}$$

where $J(x)$ is the cost function. The goal of the optimization problem is to find the optimal solution of x , which is denoted by $\hat{x}$. There are two main classifications of solutions when dealing with optimization problems, local and global. The desired solution is the global solution, which is the overall minimum point of the cost function. However, optimization solvers may settle on a local minimum, which may not necessarily be the global minimum.

In order to understand this distinction further, consider an instance of an optimization problem, which is a pair (X, J) where X is the domain of feasible points and J is the cost function from before [31]. The cost function can be thought of as a mapping,

$$J : X \rightarrow \mathbb{R}^1 \tag{2.3.13}$$

The problem then is to find an $x^* \in X$ such that

$$J(x^*) \leq J(y) \quad \forall \quad y \in X \tag{2.3.14}$$

Such a point x^* is called a globally optimal solution. Now consider that given an optimization problem with instance (X, J) , a neighborhood is defined as a mapping,

$$N : X \rightarrow Z^X \tag{2.3.15}$$

If a neighborhood is a feasible solution $c \in X$, it is then called locally optimal with respect to N ,

$$J(c) \leq J(d) \quad \forall \quad d \in N(c) \tag{2.3.16}$$

Now, given that an optimization problem with a feasible set X has a neighborhood N , if whenever $c \in X$ is locally optimal with respect to N it is also globally optimal, and the neighborhood is then defined as exact. Next consider two given points $x, y \in \mathbb{R}^n$, a convex combination of them is any point of the form,

$$z = \lambda x + (1 - \lambda)y, \quad \lambda \in \mathbb{R}^1 \text{ and } 0 \leq \lambda \leq 1 \quad (2.3.17)$$

The set $S \subseteq \mathbb{R}^n$ may then be defined as convex if it contains all convex combinations of pairs of points $x, y \in S$. An important class of optimization problems concerns the minimization of a convex function on a convex set. These type of problems are convenient in that local optima are global [31]. For all other problems, this claim cannot be explicitly made. Therefore, an optimization routine that does not minimize a convex function may settle toward the local minimum rather than the desired global one.

There exist two measures to define how close an estimate is to being optimal called the Conditions for Optimality [32]. The first order condition for optimality, which is necessary in the mathematical sense, states that the optimal point exists where gradient of the objective function is zero,

$$\nabla J(\bar{x}) = 0 \quad (2.3.18)$$

where $\bar{x}$ is the optimal solution. The necessary condition $\nabla J(\bar{x}) = 0$ is also sufficient for $\bar{x}$ to be a global minimum if f is pseudoconvex at $\bar{x}$. The term pseudoconvex implies that the objective function behaves like a convex function with respect to finding a local minimum, but it is not actually convex. A real-valued differentiable function J defined on a nonempty convex open set $X \in \mathbb{R}^n$ is pseudoconvex if $\forall x, y \in X$ such that $\nabla J(x)(y - x) \geq 0$ there exists $J(y) \geq J(x)$ [33]. The second condition requires first that the gradient of the cost function is zero and the Hessian $\mathbf{H} = \nabla^2 J(x)$ of the cost function be positive definite, $\mathbf{x}^T \mathbf{H} \mathbf{x} > 0$. It is then mathematically sufficient if there is an $\alpha > 0$ such that,

$$J(x) \geq J(\bar{x}) + \alpha \|x - \bar{x}\|^2 \quad (2.3.19)$$

for all x near $\bar{x}$.

These properties of the objective function highlight the possibility that the global optimum may not be found in every instance of an optimization problem. By increasing the complexity of the optimization algorithm, the odds of converging to the global optimum also increase. The Levenberg-Marquardt algorithm is another optimization technique that attempts to minimize a cost function, but with added complexity.

2.3.3 Newton's Method in Optimization

In calculus, Newton's method attempts to find the roots of a differentiable function $f(x)$. In the sense of an optimization problem, Newton's method (also called the Newton-Raphson method) is a gradient-based method that attempts to find the roots of a single nonlinear equation [30]. The function f must be twice-differentiable, i.e. $f'(x)$ and $f''(x)$ exist. The theorem behind Newton's method is as follows [34]

Theorem: Assume that $f \in C^2[a, b]$ and there exists $p \in [a, b]$ where $f(x) = 0$. Then, if $f'(x) \neq 0$, there exists a $\delta > 0$ such that the k^{th} iteration of x may be defined as,

$$x_k = g(x_{k-1}) = x_{k-1} - \frac{f(x_{k-1})}{f'(x_{k-1})} \quad (2.3.20)$$

for $k = 1, 2, \dots$, and it will converge to x for any initial approximation $x_0 \in [x - \delta, x + \delta]$. An expression for g may be defined as,

$$g = x - \frac{f(x)}{f'(x)} \quad (2.3.21)$$

It is apparent from Equation 2.3.21 that Newton's method will encounter a pitfall when the derivative of the function evaluated at any iteration is zero, $f'(x_{k-1}) = 0$. However, it should be noted that in this case, $f(x_{k-1})$ is an acceptable approximation of the solution. In order to work around this pitfall, Newton's method is also linearized to second order using a Taylor series's expansion about a point x_k ,

$$f(x) \approx f(x_k + \Delta x) \approx f(x_k) + f'(x_k)\Delta x + \frac{1}{2}f''(x_k)\Delta x^2 \quad (2.3.22)$$

which is then solved for Δx such that $f(x + \Delta x)$ is minimized. This is done by setting the derivative with respect to Δx equal to 0,

$$df = (f(x_k) + f'(x_k)\Delta x + \frac{1}{2}f''(x_k)\Delta x^2)d\Delta x = 0 \quad (2.3.23)$$

which yields

$$f'(x_k) + f''(x_k)\Delta x = 0 \quad (2.3.24)$$

or,

$$\Delta x = -\frac{f'(x_k)}{f''(x_k)} \quad (2.3.25)$$

Using this, the updated expression for Newton's method is,

$$x_k = x_{k-1} - \frac{f'(x_{k-1})}{f''(x_{k-1})} \quad (2.3.26)$$

In this way, when $f'(x) = 0$, $x_k = x_{k-1}$ will converge to x . Equation 2.3.26 may also be expressed in vector notation,

$$\mathbf{x}_k = \mathbf{x}_{k-1} - [\mathbf{H}f(x_{k-1})]^{-1} \nabla f(x_{k-1}) \quad (2.3.27)$$

where $\mathbf{H}f(x_{k-1}) = \nabla^2 f(x_{k-1})$ is the Hessian of the function evaluated at $\mathbf{x}_{k-1}$. The rate of convergence of Newton's method may be defined by the following equation. Suppose the sequence converges to a number L ,

$$\lim_{k \rightarrow \infty} \frac{|x_{k+1} - L|^N}{|x_k - L|} = \mu \quad (2.3.28)$$

then μ is the rate of convergence and N is the order of convergence (i.e $N=2$ corresponds to quadratic convergence). Newton's method converges very fast, especially when the initial guess is near the solution. However, computing the Hessian in higher order systems can be computationally expensive [35]. This is assuming the function being evaluated is twice differentiable to start with. If this is not the case, and the properties of Newton's method are desired in solving the function, the Quasi-Newton method may be used instead.

2.3.4 Quasi-Newton Method

A quasi-Newton method is one where the Jacobian of the function to be optimized is approximated instead of being analytically determined. There are several algorithms that are quasi-Newton in nature including the Davidon-Fletcher-Powell (DFP) [36, 37] and the Broydon-Fletcher-Goldfarb-Shanno (BFGS) [38] methods.

The DFP method was first presented by Davidon in 1959 and later updated by Fletcher and Powell in 1963. It belongs to the variable metric method family, and it has the property of quadratic convergence. Its improvement over a steepest descent method is due to the introduction of the Hessian, $\mathbf{H}$, which is an $n \times n$ matrix that contains information from all previous iterations. It is then updated at each

iteration to define the search direction. However, in order to converge the Hessian must be positive definite over the entire sequence. The DFP method is effective at solving for an optimal estimate, but it was superseded by the BFGS method.

A popular quasi-Newton method of solving nonlinear equations is the BFGS method, which was independently derived by Broydon [39], Fletcher [40], Goldfarb [41] and Shanno [42] in 1970. It converges directly to the Hessian, where the DFP method converges to the inverse of the Hessian [30]. The quasi-Newton condition imposed when updating at each iteration is $\mathbf{H}_k = \nabla f(x_k) - \nabla f(x_{k-1})$, and the Hessian is updated by adding two rank-one updates to ultimately provide a rank-two updated matrix,

$$\mathbf{H}_k = \mathbf{H}_{k-1} + \mathbf{U}_{k-1} + \mathbf{V}_{k-1} \quad (2.3.29)$$

where $\mathbf{U}_{k-1}$ and $\mathbf{V}_{k-1}$ are the matrices associated with the rank-one updates. The algorithm to the BFGS method is as follows [30].

1. Define an initial guess $\mathbf{x}_0$ and Hessian approximation $\mathbf{H}_0$
2. Determine the search direction $\mathbf{s}_k$ by solving the equation $\mathbf{H}_{k-1}\mathbf{s}_{k-1} = -\nabla f(\mathbf{x}_{k-1})$
3. Determine the step size α_{k-1} by minimizing $f(\mathbf{k})$ and then update $\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_{k-1}\mathbf{s}_{k-1}$
4. Set $\mathbf{p}_{k-1} = \alpha_{k-1}\mathbf{s}_{k-1}$
5. Define $\mathbf{y}_{k-1} = \nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1})$
6. Update the Hessian using the equation,

$$\mathbf{H}_k = \mathbf{H}_{k-1} + \frac{\mathbf{y}_{k-1}\mathbf{y}_{k-1}^T}{\mathbf{y}_{k-1}^T\mathbf{p}_{k-1}} + \frac{\mathbf{H}_{k-1}\mathbf{p}_{k-1}\mathbf{p}_{k-1}^T\mathbf{H}_{k-1}}{\mathbf{p}_{k-1}^T\mathbf{H}_{k-1}\mathbf{p}_{k-1}} \quad (2.3.30)$$

7. Continue until some convergence condition is met, usually $\|\nabla f(\mathbf{x}_k)\| < tol$

While quasi-Newton methods are more effective if the gradient or Hessian of the function being evaluated may not be computed directly, there exist other algorithms that may be better suited for this type of problem. The Levenberg-Marquardt algorithm is an optimization routine that was designed to minimize a least-squares problem which are more useful in solving nonlinear problems such as determining spacecraft attitude.

2.3.5 Levenberg-Marquardt Algorithm

Before the Levenberg-Marquardt (LM) algorithm was developed, least squares estimation of nonlinear parameters was performed using one of two methods. The first method involved expanding the model as a Taylor series and calculating the parameters at each iteration. This often fell victim to divergence of successive iterations, however. The second method involved using the method of steepest-descent to determine estimates to the nonlinear equation, which unfortunately fell victim to very slow convergence. Levenberg in 1944 and Marquardt in 1963 presented a maximum neighborhood method that performs an optimal interpolation between the Taylor series and gradient methods [43, 44, 45].

When applying the LM algorithm to a nonlinear least squares data-fitting problem, the model of the curve to be fitted to the data may be represented as,

$$\begin{aligned}\mathbf{y} &= f(x_1, x_2, \dots, x_m, b_1, b_2, \dots, b_k) \\ &= f(\mathbf{x}, \mathbf{b})\end{aligned}\tag{2.3.31}$$

where $x_1, x_2, \dots, x_m$ are independent variables and $b_1, b_2, \dots, b_k$ are the variables to be estimated. The expected value of the measurements can be expressed by using the equation $E(y) = \hat{\mathbf{y}}$,

$$\hat{\mathbf{y}} = E[f(\mathbf{x}, \mathbf{b})]\tag{2.3.32}$$

Now let the data points be represented by

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_n \end{bmatrix}\tag{2.3.33}$$

The problem then is to compute the estimates of the parameters that will minimize

$$J = \sum_{i=1}^n \left\| f_i(\mathbf{x}, \mathbf{b}) - \hat{f}_i(\mathbf{x}, \mathbf{b}) \right\|^2\tag{2.3.34}$$

It should be noted that when f is linear, the contours of J are ellipsoids, and if f is nonlinear, the contours are distorted according to the severity of the nonlinearity. Near the minimum of J however, the contours of nonlinear models are nearly elliptical to first order. Typically, the minimum of the contour surface of J lies at the bottom of a long curving trough due to it being attenuated in some direction

and elongated in others[44]. The theoretical basis of the algorithm is described by Marquardt using the following theorem

Theorem: Let $\lambda \geq 0$ be arbitrary and let $\boldsymbol{\delta}_0$ satisfy the equation

$$(\mathbf{A} + \lambda \mathbf{I})\boldsymbol{\delta}_0 = \mathbf{g} \quad (2.3.35)$$

where

$$\mathbf{A} = \mathbf{P}^T \mathbf{P} = \left(\frac{dJ_i}{db_j} \right)^T \left(\frac{dJ_i}{db_j} \right) \quad (2.3.36)$$

and

$$\mathbf{g} = \mathbf{P}^T(\mathbf{y} - f(x, \mathbf{b})) \quad (2.3.37)$$

Then $\boldsymbol{\delta}_0$ minimizes $E[J]$ on the sphere whose radius $\|\boldsymbol{\delta}\|$ satisfies $\|\boldsymbol{\delta}\|^2 = \|\boldsymbol{\delta}_0\|^2$. Using this theorem, the solution for the LM algorithm can be derived by considering the objective function from Eq. 2.3.34. In each iteration, $\mathbf{b}$ can be replaced by $\mathbf{b} + \delta$. To solve for δ , the function $f(x_i, \mathbf{b} + \delta)$ is approximated to 1st order,

$$f(x_i, \mathbf{b} + \delta) \approx f(x_i, \mathbf{b}) + \mathbf{J}_i \delta \quad (2.3.38)$$

where

$$\mathbf{J}_i = \frac{df(x_i, \mathbf{b})}{d\mathbf{b}} \quad (2.3.39)$$

is the i^{th} element of the Jacobian of f . The objective function may then be rewritten in vector form,

$$J(\mathbf{b} + \delta) \approx \|\mathbf{y} - f(x, \mathbf{b}) - \mathbf{J}\delta\|^2 \quad (2.3.40)$$

Considering the theorem from earlier, the objective function is subject to the constraint $\|\delta\|^2 = \|\delta_0\|^2$. By using Lagrange multipliers, the final form of the objective function is,

$$J(\mathbf{b} + \delta) \approx \|\mathbf{y} - f(x, \mathbf{b}) - \mathbf{J}\delta\|^2 + \lambda(\|\delta\|^2 - \|\delta_0\|^2) \quad (2.3.41)$$

The constrained optimization problem may then be solved for by taking the first differential of J with

respect to δ and setting it equal to 0,

$$dJ = [-\mathbf{y}^T \mathbf{J} + f(x, \mathbf{b})^T \mathbf{J} + \delta^T \mathbf{J}^T \mathbf{J} + \lambda \delta^T \delta] d\delta = 0 \quad (2.3.42)$$

Since $d\delta \neq 0$, the coefficient must be zero,

$$0 = -\mathbf{J}^T \mathbf{y} + \mathbf{J}^T f(x, \mathbf{b}) + \mathbf{J}^T \mathbf{J} \delta + \lambda \delta \quad (2.3.43)$$

Finally,

$$(\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I}) \delta = \mathbf{J}^T (\mathbf{y} - f(x, \mathbf{b})) \quad (2.3.44)$$

However, it should be noted that when λ gets large, $\delta \rightarrow 0$, and the convergence slows down significantly. Marquardt noticed this and expressed the $\lambda \mathbf{I}$ term using the diagonal components of the matrix $\mathbf{J}^T \mathbf{J}$. The final expression for the Levenberg-Marquardt algorithm is,

$$(\mathbf{J}^T \mathbf{J} + \lambda \text{diag}(\mathbf{J}^T \mathbf{J})) \delta = \mathbf{J}^T (\mathbf{y} - f(x, \mathbf{b})) \quad (2.3.45)$$

It can also be noted that when $\lambda = 0$ this expression is identical to the normal equations. Next, a numerical strategy was developed by Levenberg and is presented by the following procedure,

1. Let $\nu > 1$ and $\lambda^{(0)}$ be some value greater than 0
2. Compute $J(\lambda^{(r-1)})$ and $\Phi(\lambda^{(r-1)})/\nu$
 - (a) If $J(\lambda^{(r-1)})/\nu \leq J^{(r)}$, let $\lambda^{(r)} = \lambda^{(r-1)}/\nu$
 - (b) If $J(\lambda^{(r-1)})/\nu > J^{(r)}$, and $J(\lambda^{(r-1)}) \leq J^{(r)}$, let $\lambda^{(r)} = \lambda^{(r-1)}$
 - (c) If $J(\lambda^{(r-1)})/\nu > J^{(r)}$, and $J(\lambda^{(r-1)}) > J^{(r)}$, increase λ by successively multiplying by ν until for some smallest w , $J(\lambda^{(r-1)}\nu^w) \leq J^{(r)}$.
3. Let $\lambda^{(r)} = \lambda^{(r-1)}\nu^w$

This process is repeated until the value of the objective function is small enough or there is very little change between iterations. Using this algorithm, a feasible neighborhood will always be obtained [44]. The Levenberg-Marquardt algorithm provides a method that combines the benefits of using Taylor Series expansions and gradient methods, but there exist another class of optimization routines that are designed to find global optima. One popular global optimization routine is the Simulated Annealing algorithm.

2.3.6 Simulated Annealing Algorithm

By imitating an annealing process, where metals are heat treated and cooled in a controlled fashion to acquire a desired crystalline structure, the Simulated Annealing (SA) algorithm determines an optimal estimate recursively like other optimization techniques. The difference between SA and other techniques, however, is that instead of accepting the first minimum it converges to, the algorithm will force the ‘temperature’ being minimized back up according to an annealing schedule. By doing this, if the solver has settled on a local minimum, it has the chance to get out of the neighborhood of the local minimum in hopes of converging to the global one. This property is derived from the Boltzmann-Gibbs probability distribution of statistical mechanics, where the probability of a system having a particular energy state is [7],

$$P(\text{energy state} = x) = c_T \exp\left(-\frac{x}{c_b T}\right) \quad (2.3.46)$$

where $c_T > 0$ is a normalizing constant, $c_b > 0$ is known as the Boltzmann constant, and T is the temperature of the system. It should be noted that even at a low temperature, there is a possibility of reaching a higher energy state, implying that it is still possible to escape a local minimum in favor of a global one. The probability of a system going to a new energy state is given by the Metropolis criterion [46],

$$P(\text{new state}) = \exp\left(-\frac{E_{\text{new}} - E_{\text{old}}}{c_b T}\right) \quad (2.3.47)$$

After a large number of iterations, the system will settle to an equilibrium state that is governed by Eq. 2.3.46 and predicted by a fixed temperature, T . Using the Metropolis criterion, Kirkpatrick presented the SA algorithm by making a connection between statistical and combinatorial optimization.

There are a variety of algorithms in the literature, [7, 47, 48, 49] with difference between them being the annealing schedules used. In general, they all follow the same process that is defined in Figure 2.10. First an initial temperature and state estimate are supplied and the objective function is evaluated. Next, a randomly determined value of the state estimate is found and the objective function is again evaluated. The two values are then compared using the Metropolis criterion from Eq. 2.3.47, and the new estimate is accepted if the change between them $\delta < 0$. Steps 1 and 2 are repeated for some period until the budget of function evaluations allocated for T has been used, or the system reaches a state of equilibrium. The value for T is then lowered according to the annealing schedule, and the process is repeated. The final value of the estimate is taken after the entire function evaluation budget has been

used, or a condition of convergence has been met.

Figure 2.10: Process flowchart of the Simulated Annealing algorithm [7]

While global optimization routines are very effective in finding the global optimum, solution spaces that are multi-modal in nature can still lead to convergence to the wrong optimum. A popular technique in optimization, specifically path planning, involves analyzing the entire solution space before running an optimization routine to determine initial guesses that would likely yield an optimum based off of initial cost function calculations. One such technique is the use of a grid search algorithm, which will be discussed in the next section.

2.3.7 Grid Search Technique

Grid search techniques are a type of exhaustive search method that are popular in machine learning, position estimation, and path planning algorithms [50, 51, 52, 53]. They are often employed to search multidimensional spaces by dividing each plane in the space up into a grid, and evaluating each point on

the grid to determine the optimal solution. An example of this can be seen in Figure 2.11. Heuristics may be applied in cases where the solution space is large by dividing the search space into smaller candidate subspaces and evaluating the best candidate that minimizes a cost function [54]. These candidate ‘guesses’ can then be supplied to the optimization routine, and the overall process significantly decreases the required computation time compared to searching the entire solution space.

Figure 2.11: Surface plot of the x-axis solution space associated with the Big Dipper and 10 seconds between measurements. A grid is superimposed over the solution space to simulate candidate bias guess subspaces.

From here, the best candidate guesses can be supplied to any optimization routine. Combining a grid search technique with one of the local optimization routines acts similarly to a global optimization routine, since the entire solution space is searched. The advantage to doing this is that once the candidate initial guesses have been determined, the optimization process only takes a fraction of the number of iterations it would have taken if using a global optimization technique. This concept will be explored further in Chapter 3.

The previously discussed optimization techniques are classified as batch estimation techniques since they are applied once all of the data has been collected. Another technique that can be used is the application of sequential filters. One in particular, the Multiplicative Extended Kalman Filter (MEKF), will be discussed in the next section.

2.3.8 Multiplicative Extended Kalman Filter

The Kalman filter [14, 55], which was developed in 1960, is now the most common sequential filter used in state estimation. It consists of two primary steps: the propagation of a predefined dynamic model and update step through the use of sensor measurements. Various adaptations of the Kalman filter exist such as the Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF) and Multiplicative EKF (MEKF) [56, 57, 58]. The MEKF is commonly used in spacecraft attitude estimation. Attitude may be represented in a number of ways, with the most popular representations being Euler angles and quaternions. Euler angles are an intuitive way to think about 3D rotations. However, they are vulnerable to singularities and exhibit coupled nonlinear behavior that makes them a poor choice for navigation filter states. Quaternions are often used in spacecraft attitude estimation because they have no singularity. However, because quaternions are subject to a normalization constraint, $\mathbf{q}^T \mathbf{q} = 1$, the full quaternion cannot be a state in a standard EKF.

The use of a Multiplicative Extended Kalman Filter (MEKF) solves this problem by computing the multiplicative error quaternion using quaternion multiplication,

$$\delta \mathbf{q} = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1} \quad (2.3.48)$$

where $\hat{\mathbf{q}}$ represents the attitude quaternion estimate. To begin the filtering process for a system with gyros that is sequentially processing vector measurements (e.g. the LOS scope sightings of interest here), an initial attitude $\mathbf{q}_0$, bias $\mathbf{b}_0$, and covariance $\mathbf{P}_0$ estimate must be provided.

$$\begin{aligned} \hat{\mathbf{q}}(t_0) &= \mathbf{q}_0 \\ \boldsymbol{\beta}(t_0) &= \mathbf{b}_0 \\ \mathbf{P}(t_0) &= \mathbf{P}_0 \end{aligned} \quad (2.3.49)$$

From this, a state vector is defined as

$$\hat{\mathbf{x}}_0 = \begin{bmatrix} \hat{\boldsymbol{\alpha}}_0 \\ \hat{\boldsymbol{\beta}}_0 \end{bmatrix} \quad (2.3.50)$$

where $\boldsymbol{\alpha}$ is the three parameter angle vector that describes the rotation from the current estimate to truth. That is,

$$\mathbf{q} = \mathbf{q}(\boldsymbol{\alpha}) \otimes \hat{\mathbf{q}} \quad (2.3.51)$$

The value for $\boldsymbol{\alpha}$ is reset after each measurement update. Therefore, by definition, the *a priori* value for the attitude error state is always $\hat{\boldsymbol{\alpha}}_0^- = \mathbf{0}_{3 \times 1}$. When a LOS vector measurement is received, the filter calculates the Kalman gain, $\mathbf{K}_k$ and measurement sensitivity matrix, $\mathbf{H}_k$

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T (\hat{\mathbf{x}}_k^-) [\mathbf{H}_k (\hat{\mathbf{x}}_k^-) \mathbf{P}_k^- \mathbf{H}_k^T (\hat{\mathbf{x}}_k^-) + \mathbf{R}]^{-1} \\ \mathbf{H}_k (\hat{\mathbf{x}}_k^-) &= \begin{bmatrix} [\mathbf{A}(\hat{\mathbf{q}}^-) \mathbf{r}_1 \times] & 0_{3 \times 3} \\ \vdots & \vdots \\ [\mathbf{A}(\hat{\mathbf{q}}^-) \mathbf{r}_n \times] & 0_{3 \times 3} \end{bmatrix}_{t_k} \end{aligned} \quad (2.3.52)$$

where t_k represents the time at the k^{th} star LOS measurement. The filter then updates the attitude, and bias estimates by calculating the residual between the measured ($\tilde{\mathbf{y}}_k$) and predicted ($h_k(\hat{\mathbf{x}}_k^-)$) LOS vectors, and scaling it by the Kalman gain. Covariance $\mathbf{R}_k$ is also updated in this step using the Kalman gain and the measurement sensitivity matrix. The update equations used are

$$\begin{aligned} \mathbf{P}_k^+ &= [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k (\hat{\mathbf{x}}_k^-)] \mathbf{P}_k^- [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k (\hat{\mathbf{x}}_k^-)] + \mathbf{K} \mathbf{R} \mathbf{K}^T \\ \delta \hat{\mathbf{x}}_k^+ &= \mathbf{K}_k [\tilde{\mathbf{y}}_k - h_k(\hat{\mathbf{x}}_k^-)] \\ \delta \hat{\mathbf{x}}_k^+ &\equiv \begin{bmatrix} \delta \hat{\boldsymbol{\alpha}}_k^{+T} & \delta \hat{\mathbf{b}}_k^{+T} \end{bmatrix}^T \\ h_k(\hat{\mathbf{x}}_k^-) &= \begin{bmatrix} \mathbf{A}(\hat{\mathbf{q}}^-) \mathbf{r}_1 \\ \vdots \\ \mathbf{A}(\hat{\mathbf{q}}^-) \mathbf{r}_n \end{bmatrix}_{t_k} \\ \hat{\mathbf{q}}_k^+ &= \mathbf{q}(\delta \hat{\boldsymbol{\alpha}}_k^+) \otimes \hat{\mathbf{q}}_k^- \\ \mathbf{b}_k^+ &= \mathbf{b}_k^- + \delta \mathbf{b}_k^+ \end{aligned} \quad (2.3.53)$$

where $\mathbf{b}_k^+$ represents the updated bias estimate at the k^{th} iteration. Note that the Joseph form of the covariance update is used to ensure numeric stability. Between measurements, the filter propagates the attitude and covariance through dead reckoning, which uses bias corrected gyro data from the previous measurement iteration.

$$\begin{aligned}
\hat{\boldsymbol{\omega}}(t) &= \tilde{\boldsymbol{\omega}}(t) - \hat{\boldsymbol{\beta}}(t) \\
\dot{\hat{\mathbf{q}}} &= \frac{1}{2}\Xi(\hat{\mathbf{q}}(t))\hat{\boldsymbol{\omega}}(t) \\
\dot{\mathbf{P}} &= \mathbf{F}(\hat{\mathbf{x}}(t), t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^T(\hat{\mathbf{x}}(t), t) + \mathbf{G}(t)\mathbf{Q}(t)\mathbf{G}^T(t) \\
\mathbf{F}(\hat{\mathbf{x}}(t), t) &= \begin{bmatrix} -[\hat{\boldsymbol{\omega}}(t) \times] & -\mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad \mathbf{G}(t) = \begin{bmatrix} -\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix}
\end{aligned} \tag{2.3.54}$$

With a good initial state vector, process model and measurement update rate, it is reasonable to expect a good state estimate. As time goes on, the covariance of the estimate decreases to a small value, indicating convergence to a good solution. This convergence, however, is heavily dependent on providing the filter with a good initial attitude state. By nature, an MEKF will not be appropriate to solve the lost in space problem. In the next chapter, the research approach and methodology of the MOAR algorithm, and the concepts behind it will be presented.

Chapter 3

Research Approach and Methodology¹

This chapter outlines the research approach in order to provide baseline knowledge of the concepts used throughout. A brief overview on the process behind the MOAR algorithm will be presented. Next, the reference system and star catalog that were used in this work will be discussed, as well as how measurements were modeled in the simulation. Sources of noise that would be experienced in a physical implementation were accounted for in the simulation and will also be analyzed. In regard to solving for gyro bias, a review of applicable estimation techniques is presented to determine the most appropriate one. Finally, attitude estimation is discussed and the overall simulation is presented.

3.1 MOAR Algorithm Overview

The Manual Optical Attitude Re-initialization (MOAR) algorithm makes use of LOS unit vectors to targeted stars and the recorded gyro measurements between them in order to solve the lost-in-space attitude determination problem. To begin the process, a crewmember predetermines a group of stars to be targeted. In this work, it is assumed that the algorithm has already identified which stars are being targeted. An attitude solution requires observations of at least two stars if the stars being targeted are known. However, a higher number of star sightings produces a more accurate attitude solution by providing better geometric observability and redundant measurement information which may be fused in a statistically optimal way. It is for this reason that the MOAR algorithm uses a minimum of five

¹A portion of this chapter is under consideration for publication in the journal *Aerospace*

stars. Once the stars have been chosen, the star pattern (i.e. the sequence of stars) is then traced using a spotting scope. When the boresight is aligned with each star, the crewmember presses a button to ‘mark’ its LOS vector and the gyro data between it and the previous star. The process is repeated for all stars, then the algorithm, seen in Figure 3.1, may be applied.

Figure 3.1: Process flowchart of the MOAR algorithm

Bias must be removed from the gyro measurements before they are used to estimate attitude. The novel approach presented here estimates gyro bias by comparing the interior star angles as found by integrating gyro measurements and from a star catalog. An optimization routine minimizes the difference between measured and catalog interior star angles by adjusting the bias and integrating the corrected gyro measurements. Importantly, because this process relies solely on interior star angles, it is completely independent of attitude estimation. Once an estimate for the bias has been found, it can be removed from the gyro data and attitude estimation can be performed by solving the Wahba problem [15]. Although there are several methods to solve Wahba’s problem, this work makes use of a method based on the Singular Value Decomposition (SVD) [29].

Because the MOAR device is constantly moving from one star sighting to another, its inertial attitude is also changing. Thus, a reference time, t_j , is chosen and attitude is computed at this time. Although the time t_j is completely arbitrary, it usually makes sense to choose this to be the time of the first or last star sighting. This choice depends on application or user preference.

3.2 The J2000 Reference System

Reference systems can be defined for celestial objects using very distant extragalactic sources that ideally should be fixed with respect to the local reference system [59]. This is not totally realizable, however, because it is impossible to measure the positions of celestial objects directly relative to faint distant sources that include optical, radio and x-ray frequencies. It is also difficult defining the ideal reference frame due to the theories of general and special relativity. In general relativity, an inertial frame was defined by Einstein as a free falling coordinate system under the influence of the local gravitational pull caused by all of the matter in the entire universe [60]. The special theory of relativity ignores these gravitational influences and defines an inertial frame in a space-time continuum. Quasi-inertial frames may be defined as regions where particles move freely without accelerations where the laws of special relativity hold with high accuracy. The Earth-Moon system in orbit around the Sun is an example of such a quasi-inertial frame, and any violations of special relativity may be corrected in the appropriate coordinate-time reference frame.

Ideally quasi-inertial frames are constructed from extragalactic sources that are assumed to have no rotational dynamics in their motion. However, sources that do have rotational components may also be used. Differences in these references systems need to be recognized, and they can be more accurately described using differences in time. The current standard epoch that is used as a reference for star catalogs and defining celestial objects' positions is [61],

$$J2000 = 2000 \text{ January } 1.5 \tag{3.2.1}$$

which corresponds to the Julian date system starting in the year 2000. Previously the Besselian system was used, and is still in use sometimes today. However, there are two major disadvantages associated with it. First, the length of a year varies slowly, and secondly the instances at the beginning of a year do not correspond to a Julian date. This is inconvenient when describing astronomical events, which is why the Julian system is more commonly used in defining celestial objects positions. The J2000 equatorial coordinate system is defined by using the Earth's mean equinox as the x-axis, then defining the y-axis by making a plane along the mean equator. The z axis is then defined by making a right-handed system around the x and y axes, which can be seen in Figure 3.2. In the equatorial coordinate system, positions of celestial objects are described using right ascension α and declination δ angles.

There are other ways to define celestial coordinates, another important one using ecliptic coordinates, which make use of the mean equinox and the ecliptic for the x-y plane. The coordinate system uses ecliptic longitude λ and ecliptic latitude β to describe positions of celestial objects. The ecliptic and

Figure 3.2: J2000 Equatorial Coordinate System, inspired by diagram from Navipedia [8]

the celestial equator are slowly moving with respect to each other due to small perturbing forces acting on the Earth. This motion slowly turns the coordinate system westward with a period of about 26,000 years; to correct for this motion, an epoch such as J2000 is required. Now that the reference system is defined, star locations may be defined data from a star catalog.

3.3 Star Catalog

Inertial star directions were retrieved from the Bright Star Catalog, which contains information on 9,096 stars represented in the J2000 coordinate system [62]. Star directions are reported in terms of right ascension, α , and declination, δ . Using these parameters, line-of-sight (LOS) unit vectors to the i -th star may be expressed in the inertial frame,

$$\mathbf{u}_{i_I} = \begin{bmatrix} \cos \delta_i \cos \alpha_i \\ \cos \delta_i \sin \alpha_i \\ \sin \delta_i \end{bmatrix} \quad (3.3.1)$$

A representation of the unit vector to each star can be seen in Figure 3.3. By assuming that the stars are significantly far away, the LOS computed from the star catalog may be treated as constant (no parallax). In this work there are two reference frames being considered: the inertial frame and the sensor frame.

Star catalog LOS data is defined in the inertial frame, and measurements taken by the MOAR device at a particular instance in time are in the sensor frame.

Figure 3.3: Star LOS vector defined by right ascension α and declination δ

3.3.1 Effect of Parallax on Apparent Star Directions

A property of star catalogs that must be addressed is parallax, or the phenomenon where the location of the observer affects the apparent location of the object being viewed. Parallax can have an impact on defining star LOS vectors by introducing errors that can be propagated through attitude estimation. An example was considered in Figure 3.4, that looks at the effect of parallax from observing Alpha Centauri, the closest star to the Solar System.

Figure 3.4: Illustration of the effect of parallax on the closest star to the Solar System, Alpha Centauri

Assuming that an observer is standing on Earth, the total change in distance of the observer over the course of a year is 2 astronomical units (AU), where 1 AU is 1.496×10^8 km. Using that as the height of a triangle and the distance from the Sun to Alpha Centuri, 4.36 light-years (ly), as the base, the angle 2θ may be found by solving,

$$\begin{aligned}\theta &= \arctan\left(\frac{1 \text{ AU}}{4.37 \text{ ly}}\right) \\ &= \arctan\left(\frac{1.5 \times 10^8 \text{ km}}{4.13 \times 10^{13} \text{ km}}\right) \\ &= 2.08 \times 10^{-4} \text{ deg}\end{aligned}\tag{3.3.2}$$

therefore,

$$2\theta = 4.162 \times 10^{-4} \text{ deg}\tag{3.3.3}$$

which corresponds to about 1.5 arcsec. Since Alpha Centuri is the closest star to the Solar System, this is the worst-case scenario for error due to parallax. Because of this, the effects of parallax may be considered to be negligible, as they were in this work. As a result, the lines of sight from the star catalog may be treated as constant. These LOS vectors from the star catalog can be used in the attitude determination process as the true star LOS data. Another potential effect on star directions that must be accounted for is stellar aberration.

3.3.2 Effect of Stellar Aberration on Apparent Star Directions

Stellar aberration, an effect of special relativity, produces an apparent motion of celestial objects. As a result, the LOS vector to a star appears to be different in two inertial frames that have a relative velocity between them. The mathematics behind this concept are explained in detail in [63]. Computing aberration effects using a Newtonian approach will yield approximately the same result as using Special Relativity (to order v/c) [63]. However, not using a Special Relativity approach can yield physical errors as large as the stellar aberration errors themselves.

To quantify these effects, two frames must be defined. The first is an inertial frame denoted by the subscript I , and the second, denoted by I' , is a frame whose axes are parallel with the first frame, but the origin is moving a velocity $\mathbf{v}$ relative to I . Because of special relativity, recall that $\mathbf{u}_{i_I} \neq \mathbf{u}_{i_{I'}}$. Since direction of a star is opposite to the direction of the velocity of the photons emitted from the star to the observer, a variable β can be defined as the velocity vector $\mathbf{v}$ of the observer normalized by the speed of

light c ,

$$\boldsymbol{\beta} = \frac{\mathbf{v}}{c} \quad (3.3.4)$$

and its magnitude is never larger than 1. The rotation vector associated with stellar aberration is given by [64]

$$\delta\boldsymbol{\theta}_{aberration} \approx \boldsymbol{\beta} \times \mathbf{u}_{i_I} \quad (3.3.5)$$

where $\mathbf{u}_{i_I}$ is any unit vector in the inertial frame. This rotation vector can be expressed as a rotation matrix,

$$\mathbf{T}_{aberration} \approx \mathbf{I}_{3 \times 3} - [\delta\boldsymbol{\theta}_{aberration} \times] \quad (3.3.6)$$

which can be used to define the apparent rotation between the observed and true LOS vectors,

$$\mathbf{u}_{i_{I'}} = \mathbf{T}_{aberration} \mathbf{u}_{i_I} \quad (3.3.7)$$

This should not be considered a physical rotation between the two frames, but rather a way to quantify the apparent rotation between the two vectors. Next, measurement LOS vectors in the sensor frame were defined.

3.4 Manual Scope Measurements

By targeting stars with the scope, the user obtains a star LOS measurement. Defining the sensor's z -axis to be along the scope boresight direction, any ideal star LOS measurement in the sensor frame is simply

$$\mathbf{u}_{i_{S_i}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.4.1)$$

That is, the i -th LOS star measurement, as expressed in the sensor frame at time t_i (S_i), is entirely along the z - direction. This is always the same, no matter which star is being targeted.

Because the sensor is moved between star sightings, the inertial orientation of the sensor is always changing. Thus, it is desirable to express all scope measurements in the sensor frame at a particular reference time, t_j (the sensor frame at t_j is denoted as S_j). This transformation simply requires the

scope measurement at time t_i be rotated into the sensor frame at time t_j . This is given by

$$\mathbf{u}_{iS_j} = \mathbf{T}_{S_j}^{S_i} \mathbf{u}_{iS_i} = \mathbf{T}_{S_j}^{S_i} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.4.2)$$

where $\mathbf{T}_{S_j}^{S_i}$ is the 3×3 rotation matrix that transforms a vector from the sensor frame at time t_i to the sensor frame at time t_j . Gyro data can be used to find this mapping between sensor frames. An illustration of this process can be seen in Figure 3.5, where 3 example stars are observed.

Figure 3.5: Example of mapping the rotation between three star sightings

3.5 IMU Measurements

Inertial Measurement Units (IMUs) are sensors that consist of an accelerometer that measures sensed acceleration and a gyroscope that measures body rotation rates. Only the gyro data is necessary for the present application. Gyroscope data were modeled using the Farrenkopf gyro model [65],

$$\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} + \mathbf{b} + \boldsymbol{\nu} \quad (3.5.1)$$

where $\tilde{\boldsymbol{\omega}}$ represents the measured (corrupted) body rate, $\boldsymbol{\omega}$ is the true body rate, $\mathbf{b}$ is the gyro rate bias, and $\boldsymbol{\nu}$ is zero mean white noise. The estimate of the body rate can be found by taking the expected value,

$$\hat{\boldsymbol{\omega}} = E[\boldsymbol{\omega}] = E[\tilde{\boldsymbol{\omega}} - \mathbf{b} - \boldsymbol{\nu}] = \tilde{\boldsymbol{\omega}} - E[\mathbf{b}] - E[\boldsymbol{\nu}] = \tilde{\boldsymbol{\omega}} - \hat{\mathbf{b}} \quad (3.5.2)$$

where $E[\cdot]$ is the expected value operator. Assuming that the change in attitude between updates in gyro data is relatively small, the rate measurements may be converted into a small rotation

$$\delta\boldsymbol{\theta} = \hat{\boldsymbol{\omega}} \Delta t \quad (3.5.3)$$

where $\delta\boldsymbol{\theta}$ is an angle vector and Δt is the time step between sequential gyro measurements. The attitude change described by this angle vector may be converted to a quaternion,

$$\delta\mathbf{q} = \begin{bmatrix} \sin\left(\frac{\|\delta\boldsymbol{\theta}\|}{2}\right)\mathbf{e}_{\boldsymbol{\theta}} \\ \cos\left(\frac{\|\delta\boldsymbol{\theta}\|}{2}\right) \end{bmatrix} \quad (3.5.4)$$

and the attitude estimate may be propagated by

$$\mathbf{q}_k = \delta\mathbf{q} \otimes \mathbf{q}_{k-1} \quad (3.5.5)$$

where $\mathbf{q}_{k-1}$ is the attitude quaternion at the previous time step, and the symbol $\otimes$ denotes quaternion multiplication given by

$$\mathbf{q} \otimes \mathbf{p} = \begin{bmatrix} q_4\mathbf{p}_v + p_4\mathbf{q}_v - \mathbf{q}_v \times \mathbf{p}_v \\ q_4p_4 - \mathbf{q}_v^T \mathbf{p}_v \end{bmatrix} \quad (3.5.6)$$

If the gyro body rates are integrated continuously from a starting time (e.g. device turn-on, first star ‘mark,’ etc.), then the integrated quaternion describes the rotation from the sensor frame at the starting time t_0 (S_0) to the sensor frame at time t_i (S_i). Recalling that the attitude quaternion may be converted into a rotation matrix, the integrated quaternion at time t_i may be used to construct the rotation matrix

$$\mathbf{T}_{S_i}^{S_0} = (q_s^2 - \mathbf{q}_v^T \mathbf{q}_v)\mathbf{I}_{3 \times 3} + 2\mathbf{q}_v \mathbf{q}_v^T - q_s[\mathbf{q}_v \times] \quad (3.5.7)$$

where the attitude quaternion is composed of vector part ($\mathbf{q}_v$) and a scalar part (q_s).

Therefore, the estimate of the rotation from the sensor frame at measurement time t_i (S_i) to the sensor frame at the reference time t_j (S_j , recalling this is the time at which the sensor’s attitude is computed) is simply

$$\mathbf{T}_{S_j}^{S_i} = \mathbf{T}_{S_j}^{S_0} \mathbf{T}_{S_0}^{S_i} = \mathbf{T}_{S_j}^{S_0} \left(\mathbf{T}_{S_i}^{S_0} \right)^T \quad (3.5.8)$$

This provides the frame transformation necessary to compute the measurements described in Eq. 3.4.2. The next step is to determine gyro bias and remove it from the system so that an accurate attitude

estimate may be found.

3.5.1 Angle Random Walk

Angle random walk (ARW), with units of $deg/\sqrt{hr}$, is one of the most common specifications for describing the performance of a gyroscope [66]. The concept of random walk is used in statistics to describe what happens when the output of a system subjected to random noise is integrated. A common example is supposing that a person starts at some reference origin and takes a set number of steps forward or backward at random, with a 50 percent chance of stepping in either direction. Over time, the person would be expected to be near the starting point, but it is possible that the person could be significantly far away from the origin. The same concept can be applied to noise in sensors; ARW specifically describes the angle deviation that will occur when the noisy gyroscope signal is integrated [67]. Because of this noise in the body angle rate measurements, attitude estimation error will increase over longer integration times. As one might infer from the units of the ARW, it can be shown that the variance grows linearly with time or that the standard deviation grows with the square root of time.

There are several ways to quantify the ARW property of a sensor, such as using Power Spectral Density (PSD) or a Fast Fourier Transform (FFT). Standard deviation (σ) of the signal may also be provided, in addition to the effective bandwidth (BW) of the sensor. Because gyro manufacturers' standards may differ, it is important to know how to find ARW based on the information at hand. What follows is a quick summary of how to convert between various noise specifications [67]:

$$\begin{aligned}\sigma_{ARW} &= \sqrt{PSD} \\ \sigma_{ARW} &= FFT \\ \sigma_{ARW} &= \sigma_v \frac{1}{\sqrt{BW}}\end{aligned}\tag{3.5.9}$$

Caution should be exercised when using ARW values to ensure that units are consistent. ARW is typically given in units of $deg/\sqrt{hr}$, but it may be needed to match a time scale on the order of seconds. There also exists an expression that relates ARW and the ARW dominated portion of the Allan variance (which will be discussed in Section 3.5.3),

$$\sigma_\omega^2(\tau) = \frac{\sigma_{ARW}^2}{\tau}\tag{3.5.10}$$

Another performance characteristic of gyros that must be accounted for is bias instability.

3.5.2 IMU Bias Instability

When a sensor is powered on, there exists a random bias that varies with each power on cycle. However, over time the bias may drift away from the initial bias value. The rate of change in bias can be modeled by the first order Gauss-Markov process,

$$\dot{\mathbf{b}} = \frac{1}{\tau}\mathbf{b} + \boldsymbol{\eta} \quad (3.5.11)$$

where τ is correlation time and $\boldsymbol{\eta}$ represents the intensity of the Gauss-Markov process, and as such drives the bias instability in the gyro. Sensors with a lower bias instability value experience less fluctuation in bias. It may be determined graphically from the graph for the Allan variance of a sensor, where it corresponds to the $\sigma_{\omega}^2(\tau)$ at the bottom of the Allan variance curve. It may also be found analytically using the expression,

$$\sigma_{\omega}^2(\tau) = \frac{2B}{\pi} \ln 2 \quad (3.5.12)$$

where B represents the bias instability and σ_{ω}^2 is the Allan variance. In this work three different grades of IMUs are considered: MEMS, tactical and navigational. MEMS IMUs are small and inexpensive but susceptible to bias instability. Tactical grade IMUs are more expensive than the MEMS variety, and with that comes a more stable bias. Navigation grade IMUs are the most expensive available and have the best bias stability characteristics. Section 3.6.8 will discuss the effect that all 3 grades of IMUs have on the MOAR algorithm.

3.5.3 Allan Variance

The Allan variance is a measure of frequency stability in sensors that analyzes a sequence of data to determine a system's noise across varying averaging time [68]. It is typically computed for gyros by analyzing data with sliding windows of varying size,

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}(t_k) = \int_{t_0}^{t_k} \boldsymbol{\omega}(\zeta) d\zeta \quad (3.5.13)$$

and if measurements are equally spaced in time, as discrete sensors are the above equation can be rewritten,

$$\boldsymbol{\theta}_k = \Delta t \sum_{i=0}^k \boldsymbol{\omega}(t_k) \quad (3.5.14)$$

Average rate can then be defined over time $\tau = m\Delta t$, and the window of selected gyro data can be represented as,

$$\bar{\omega}(\tau) = \frac{\boldsymbol{\theta}(t_k + \tau) - \boldsymbol{\theta}(t_k)}{\tau} \quad (3.5.15)$$

which can be rewritten discretely,

$$\bar{\omega}(\tau) = \frac{\boldsymbol{\theta}_{k+m} - \boldsymbol{\theta}_k}{\tau} \quad (3.5.16)$$

The definition of the Allan Variance is,

$$\sigma_{\omega}^2(\tau) = \frac{1}{2} \langle [\bar{\omega}_{k+m}(\tau) - \bar{\omega}_k(\tau)]^2 \rangle \quad (3.5.17)$$

where $\langle \cdot \rangle$ denotes the ensemble average. This can be rewritten in terms of the integrated signal,

$$\sigma_{\omega}^2(\tau) = \frac{1}{2\tau^2} \langle (\boldsymbol{\theta}_{k+2m} - 2\boldsymbol{\theta}_{k+m} \boldsymbol{\theta}_k)^2 \rangle \quad (3.5.18)$$

finally, an expression for the Allan variance in terms of the integrated signal is given by,

$$\sigma_{\omega}^2(\tau) = \frac{1}{2\tau^2(N-2m)} \sum_{k=1}^{N-2m} (\boldsymbol{\theta}_{k+2m} - 2\boldsymbol{\theta}_{k+m} \boldsymbol{\theta}_k)^2 \quad (3.5.19)$$

From here the Allan Variance can be used to describe the performance of the gyro. Graphical analysis may be performed using the plot for Allan variance as well. For example, the variance of ARW of a gyro can be found directly from the plot by determining the Allan variance at an averaging time of 1 second. Bias instability can also be found from the plateau in the plot. The ADIS16405 IMU was chosen for implementation on the MOAR prototype, and the variance of gyro ARW was found to be around $2.0^\circ/\sqrt{hr}$, or $0.033^\circ/s$ according to the datasheet [9]. Now that bias instability and Allan variance have been defined, bias can be determined more accurately.

3.6 Gyro Bias Estimation

Before the spacecraft's attitude can be estimated, gyro bias must be determined and removed from the gyro measurements. One method for estimating bias, which is completely independent of attitude estimation, is to use an optimization technique that minimizes the difference between the interior star angles measured by the sensor and the corresponding interior star angles computed from a star catalog.

3.6.1 Cost Function Definition

To begin, the optimization routine is supplied with an initial bias guess, which is then removed from the measured gyro data to arrive at corrected body rate measurements,

$$\hat{\omega} = \tilde{\omega} - \hat{\mathbf{b}} \quad (3.6.1)$$

The corrected body rate measurements may then be integrated to determine the interior star angles between each star. In order to prevent flattening of the interior star angles over time, all possible non-repeating combinations of stars are considered.

Recall that all star measurements can be mapped to a common sensor reference frame, S_j , as described in Eq. 3.4.2. The rotation matrix $\mathbf{T}_{S_j}^{S_i}$ is computed from the integrated gyro data according to Eqs. 3.5.3–3.5.8. Thus, the angle between the i -th and k -th star measurements is simply

$$\tilde{\theta}_{ik} = \cos^{-1} \left(\tilde{\mathbf{u}}_{iS_j}^T \tilde{\mathbf{u}}_{kS_j} \right) \quad (3.6.2)$$

where the tilde indicates that these are measured quantities. Likewise, the corresponding interior star angle may be computed from the star catalog LOS vectors from Eq. 3.3.1,

$$\theta_{ik} = \cos^{-1} \left(\mathbf{u}_{iI}^T \mathbf{u}_{kI} \right) \quad (3.6.3)$$

where the absence of the tilde indicates these are quantities derived from the star catalog.

The interior star angles for every unique star pair are now stored in a vector. For a set of seven stars, this would create an interior star vector of

$$\tilde{\boldsymbol{\theta}} = \begin{bmatrix} \tilde{\theta}_{12} \\ \tilde{\theta}_{13} \\ \tilde{\theta}_{14} \\ \tilde{\theta}_{15} \\ \tilde{\theta}_{16} \\ \tilde{\theta}_{17} \\ \tilde{\theta}_{23} \\ \vdots \\ \tilde{\theta}_{67} \end{bmatrix} \quad (3.6.4)$$

This vector of interior star angles is used to form cost function $J(\mathbf{b})$ that is minimized in the optimization

process,

$$\text{Minimize } J(\mathbf{b}) = \|\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}\|^2 \quad (3.6.5)$$

To solve for the optimal bias estimate, an initial guess is made and the cost function is evaluated. If the value is not smaller than a predefined threshold, the bias guess is updated. Changing the bias guess will change the corrected gyro data and the resulting integrated attitudes — which will, in turn, change the measured interior star angles. This process is repeated until either (1) the value of the cost function meets the threshold condition or (2) there is a negligible change in the value cost function from iteration to iteration. Consideration of the equations leading to the cost function in Eq. 3.6.5 reveals that the problem is both nonlinear and multimodal. Several optimization strategies were considered, and are highlighted in the following sections. A contour plot of how the solution space associated with the cost function changes with different bias values can be seen in Figure 3.6.

Figure 3.6: Solution space of the bias estimation cost function with a step size between measurements of 10 seconds.

Once an estimate for the bias has been found, it may then be subtracted from the original gyro data signal to arrive at the gyro estimate. From here, an estimate for the attitude may be computed. A flowchart of the bias estimation process can be seen in Figure 3.7. Now that a routine has been defined to estimate bias, it must be tested to determine its accuracy for use in the MOAR algorithm. There exist many types of estimation techniques such as optimization and filtering. The next section highlights a simple unconstrained local optimization routine.

3.6.2 Unconstrained Optimization Implementation

In this work, a quasi-Newton unconstrained optimization technique was tested in its effectiveness to determine gyro bias. The *fminunc* function from the Matlab Optimization Toolbox was selected as a basic minimization routine. The Broydon-Fletcher-Goldfarb-Shanno (BFGS) algorithm was used for the minimization because it does not require the gradient of the cost function to be calculated. This is ideal

Figure 3.7: Process Flowchart for the Bias Estimation Routine

for this application because an analytic expression for the gradient is not easily found.

To begin the simulation a true bias vector, $\mathbf{b} = [2.865 \ 2.865 \ 2.865]^T \text{ deg/s}$ was chosen based on 1σ values of the gyro bias from the ADIS16405 data sheet [9]. In this simulation, bias instability was not considered, since the anticipated measurement time was short. The procedure presented in the previous section was implemented with the maximum number of iterations allowed set to 2000. Next, 200 different initial bias guesses were supplied to the optimization routine, and bias estimates were determined. Noise was simulated on the gyro measurements with $\sigma_{gyro} = 0.1261 \text{ deg/s}$, and pointing error with $\sigma_{pointing} = 2.778 \times 10^{-3} \text{ deg}$ (or 10 arcsec) was added to simulated LOS measurements. Results from this simulation can be seen in Figure 3.8.

The bias estimates were plotted in 2-axes combinations, (x-y), (x-z), (y-z) to show the performance of each bias component and their relationship. It is immediately apparent that the bias estimates do not converge to the true bias; they have a magnitude on the order of 10^2 , whereas the true bias is on the order of 10^1 . The error in these estimates can be seen in the error histogram in Figure 3.8. Table 3.1 displays all possible exit conditions for the *fminunc* solver and their explanations, as well as their frequency of occurrence.

The three solver termination conditions that resulted from using *fminunc* were: the change in objective function was smaller than the set tolerance (Condition 5), the change in the bias estimate was

Figure 3.8: BFGS bias estimates using *fminunc* function in Matlab, with a red diamond denoting the true bias (left) and error histograms of the BFGS bias estimates (right)

Table 3.1: Termination conditions for *fminunc*, returned as integer

	Exit flag integer values	Percent of cases
1	Magnitude of gradient is smaller than the TolFun tolerance.	0%
2	Change in x was smaller than the TolX tolerance.	6%
3	Change in the objective function value was less than the TolFun tolerance.	0%
5	Predicted decrease in the obj function was less than the TolFun tolerance.	91%
0	Max no. of iterations/function evals exceeded Max.	3%
-1	Algorithm was terminated by the output function.	0%
-3	Objective function at current iteration went below ObjectiveLimit.	0%

smaller than the set tolerance (Condition 2) and the maximum number of iterations had been reached (Condition 0). The most common reason (91% of cases) for BFGS terminating was due to the predicted increase in the objective function being less than the function difference tolerance. This means that the change in the objective function was small enough that the solver determined a minimum had been found. However, upon further inspection it is apparent that the BFGS routine did not converge to the

global (true) minimum. With this in mind, it should be apparent that the *fminunc* solver does not accurately estimate bias, and therefore would not be used in the MOAR algorithm. The next solver that was considered was the Levenberg-Marquardt algorithm.

3.6.3 Levenberg-Marquardt Implementation

Since the BFGS quasi-Newton algorithm provided to be inaccurate in estimating bias, the Levenberg-Marquardt algorithm was investigated as a way to accurately determine gyro bias. The *lsqnonlin* function from the Matlab Optimization toolbox is a method that can be used to solve nonlinear least-squares problems. The algorithm used in the *lsqnonlin* function by default is the ‘Trust Region Reflective’, but it may be changed to the ‘Levenberg-Marquardt’ algorithm, as was done in this case. The LM algorithm used in this work can be outlined by the following steps,

1. Pick an initial guess for the bias, $\mathbf{b}_{guess}$
2. Subtract bias from the gyro data $\hat{\boldsymbol{\omega}} = \tilde{\boldsymbol{\omega}} - \mathbf{b}_{guess}$ and determine interior star angles for each star combination
3. Compute the Jacobian for the system, $\mathbf{J} = \frac{df}{d\mathbf{b}}$ (Matlab does this numerically if one is not supplied)
4. Solve the following system for $\boldsymbol{\delta}$

$$[\mathbf{J}^T \mathbf{J} + \lambda \text{diag}(\mathbf{J}^T \mathbf{J})] \boldsymbol{\delta} = \mathbf{J}^T [\tilde{\mathbf{y}} - f(x, \mathbf{b})] \quad (3.6.6)$$

5. Compute $\gamma = \mathbf{b} + \boldsymbol{\delta}$ and compare the residuals of $[\tilde{\mathbf{y}} - f(x, \mathbf{b})]$ and $[\tilde{\mathbf{y}} - f(x, \gamma)]$, and then do one of the following:
 - (a) If γ produces a reduction in the residual, then it is an improvement over the last iteration so $\mathbf{b}$ is set equal to γ . The value for λ is then reduced by an order of magnitude, $\lambda^{k+1} = \lambda/10$.
 - (b) If γ produces an increase in the residual, then it is a step in the wrong direction over the last iteration. The value for λ is then increased by an order of magnitude, $\lambda^{k+1} = 10\lambda$ and Step 4 is attempted again.
6. Continue until the residual of $\|\tilde{\mathbf{y}} - f(x, \mathbf{b})\| < tol$ or a maximum number of iterations has been reached.

Like in the BFGS simulation, the true bias vector was set to $\mathbf{b} = [2.865 \ 2.865 \ 2.865]^T \text{ deg/s}$, the same 200 randomly generated bias guesses from before were supplied to the solver, and the maximum iteration

limit was also set to 2000. Noise was again simulated on the gyro measurements with $\sigma_{gyro} = 0.1261 \text{ deg/s}$, and pointing error with $\sigma_{pointing} = 2.778 \times 10^{-3} \text{ deg}$ (or 10 arcsec) was added to simulated LOS measurements. Results from the second simulation can be seen in Figure 3.9.

Figure 3.9: LM bias estimates using *lsqnonlin* function in Matlab, with a red diamond denoting the true bias (left) and error histogram of the LM bias estimates (right)

The bias estimates were again plotted in 2-axes combinations, (x-y), (x-z), (y-z) to show the performance of each bias component and their relationship. They showed improvement in convergence over the unconstrained optimization case. However, the bias error had a magnitude on the order of 10^2 , showing poor bias estimation still. Table 3.2 displays all possible exit conditions for LM and their explanations, as well as their percentage of how often they occurred.

This shows that the majority of bias estimates determined by *lsqnonlin* were due to the change in residual between iterations meeting the tolerance. The rest were due to the solver terminating at the maximum number of iterations allowed. It is possible to increase the number of maximum iterations further, however, that could have adverse effects if the estimates in fact are diverging when the maximum

Table 3.2: Termination conditions for *lsqnonlin*, returned as integer

Exit flag integer values		Percent of cases
1	Function converged to a solution x.	0%
2	Change in x was less than the specified tolerance.	0%
3	Change in the residual was less than the specified tolerance.	69.5%
4	Magnitude of search direction was smaller than the specified tolerance.	0%
0	No. of iterations exceeded MaxIter or no. of function evals exceeded Max.	30.5%
-1	Output function terminated the algorithm.	0%
-2	Problem is infeasible: the bounds lb and ub are inconsistent.	0%
-4	Could not sufficiently decrease the residual along the current search direction.	0%

number of iterations are reached. It would make more sense to use an optimization technique that was designed to have a high number of iterations and converge to the global optimum, such as Simulated Annealing.

3.6.4 Simulated Annealing Implementation

With poor results using BFGS and LM, the global optimization technique Simulated Annealing was considered for implementation in the MOAR algorithm. Because of its willingness to back out of a local minimum in favor of finding a global one, better results were expected. Like in the BFGS and LM simulations, the true bias vector was set to $\mathbf{b} = [2.865 \ 2.865 \ 2.865]^T \text{ deg/s}$, the same 200 initial bias conditions from before were supplied to the SA simulation using the function *simulannealbnd* in Matlab. Noise was again simulated on the gyro measurements with $\sigma_{gyro} = 0.1261 \text{ deg/s}$, and pointing error with $\sigma_{pointing} = 2.778 \times 10^{-3} \text{ deg}$ (or 10 arcsec) was added to simulated LOS measurements. Bounds of $[-5.73 \ 5.73] \text{ deg}$ were applied for all estimate 3 components to keep any diverging estimates near the truth so data analysis would be visually meaningful. The resulting bias estimates can be seen in Figure 3.10.

The results show that SA had better convergence than the LM estimates, especially in the x-y components, with bias error values on the order of 10^{-1} . However, the z-component estimates did not tightly converge to one value; instead, they converged to a line about the true x and y-axis estimates. At this point it became apparent that the z-axis is weakly observable, and this issue will be investigated further in Section 3.6.7. After evaluating SA, a reliable method to estimate bias had not yet been found. Recalling that local optimization techniques are fast and accurate given that the initial bias guess is in the neighborhood of the truth, a BFGS-grid search algorithm was evaluated for use in the MOAR algorithm.

Figure 3.10: Simulated Annealing bias estimates using *simulannealbnd* function in Matlab, with a red diamond denoting the true bias (left) and error histogram of the SA bias estimates (right)

3.6.5 Optimization Using a Grid Search Algorithm and BFGS

At this point 3 different optimization routines have been tested in their effectiveness of estimating bias. Due to the complexity of the problem, BFGS and LM could not converge to the truth reliably, given any initial bias estimate. Despite SA being a global optimization technique that searches a large solution space, reliable estimates for the z-component of the bias could not be found because the z-axis is weakly observable in the problem.

Next, a grid search was considered to search the entire solution space for bias guess candidates that would provide a starting point in the neighborhood of the global minimum. A 3D search space was defined by setting reasonable boundaries to $\pm 3\sigma$ of the sensor bias provided by the data sheet. Statistically, within these boundaries there is a 99% probability of finding the true bias. From here, an even sampling may be applied to each axis to create a grid pattern. In this work, the grid was defined as having 17 nodes on each axis, which resulted in 4,913 different initial bias guesses. Each of these

guesses was evaluated in the grid search algorithm and the top 5 candidates were selected to be used in the BFGS optimization routine. Of these 5, the one with the lowest cost function value was selected as the optimal bias estimate. Like the previous 3 cases, a Monte Carlo simulation consisting of 200 iterations was performed and the results can be seen in Figure 3.11. Noise was again simulated on the gyro measurements with $\sigma_{gyro} = 0.1261 \text{ deg/s}$, and pointing error with $\sigma_{pointing} = 2.778 \times 10^{-3} \text{ deg}$ (or 10 arcsec) was added to simulated LOS measurements.

Figure 3.11: BFGS-grid bias estimates (left) and error histogram of bias estimates found using the BFGS-grid algorithm (right)

Final bias estimates from the BFGS-grid method show the best convergence to the true bias yet, especially for the x-y pairs. The bias error for the x and y components was on the order of 10^{-2} deg , and the z component error was on the order of 10^{-1} deg , which is an order of magnitude improvement over using SA. There is still a slight amount of error associated with the z axis, which was expected, but it was the lowest error of any of the methods presented for bias estimation.

With various optimization routines having been tested, a different approach was analyzed that is

popular in vehicle attitude estimation, using a Multiplicative Extended Kalman Filter.

3.6.6 Multiplicative EKF Implementation

The previous sections highlighted different ways to estimate bias using various optimization techniques in a batch estimation capacity. A popular estimation technique in aerospace systems is the use of a Multiplicative Extended Kalman Filter (MEKF), which propagates the expected state value until a measurement is received, then the state is updated and the process is repeated with each sequential measurement. In order to be an effective estimator, the MEKF must have some *a priori* knowledge of the state(s) being estimated, in this case bias and attitude, otherwise the filter can diverge.

Two Monte Carlo simulations were performed involving the MEKF in this work. The first simulation operated on the assumption that there was *a priori* knowledge of the bias and attitude states. The initial bias guess was the true bias vector with added noise that had a sigma value of 0.286 deg/s , while the initial attitude guess was the true attitude quaternion with multiplicative noise that had a sigma value of 5 deg . A total of 200 cases were simulated, and the results of this simulation may be seen in Figure 3.12.

Figure 3.12: Bias (left) and attitude (right) error plots from the MEKF Monte Carlo simulation with good *a priori* knowledge of bias and attitude. Bias and attitude estimates over time for all 200 Monte Carlo results are in gray with 3σ covariance bounds in black.

The plots from this first simulation show that the MEKF was effective in estimating both bias and attitude. Error values for both decreased over time, and were within the 3σ error bounds, which are

represented by the black lines overlaying the data. The star sighting times, which occur every 10 seconds, are evident and correspond to the discrete reduction of both estimate errors and filter covariance. The bias error values were exceptional, starting around $\pm 1 \text{ deg/s}$ and decreasing to below $\pm 0.5 \text{ deg/s}$ by the last measurements. Attitude showed excellent convergence for the x and y axes, as the error values decrease drastically over time. As expected, the z axis did not converge as tightly to the truth due to the weakly observable nature of the z -axis. Additionally, the analytic MEKF covariance accurately describes the uncertainty in both the attitude and bias estimates.

A second Monte Carlo simulation was performed that assumed there was very poor *a priori* knowledge of bias or attitude. The initial bias guess for this case was the true bias vector with added noise that had a sigma value of 1 deg/s , while the initial attitude guess was the true attitude quaternion with multiplicative noise with a sigma value of 180 deg . Again, a total of 200 cases were simulated, and the results may be seen in Figure 3.13. The initial bias guesses were all set to zero for this case, as the grid generates candidate guesses to be evaluated.

Figure 3.13: Bias (left) and attitude (right) error plots from the MEKF Monte Carlo simulation with very poor *a priori* knowledge of bias and attitude. Bias and attitude estimates over time for all 200 Monte Carlo results are in gray with 3σ covariance bounds in black.

As expected, with a poor initial guess of bias and attitude, the MEKF did not converge on the optimal estimates. This is because the MEKF update is based on a linearization about the *a priori* estimate. If the *a priori* estimate is not close to the truth, then the linearization will be invalid, and the filter will not function properly. This is especially evident in the 3σ covariance lines, with bias covariance

barely changing with time, and the attitude covariance drastically and frequently changing over time without converging. The results from these plots illustrate the need for good initial bias and attitude guesses. Since the lost-in-space problem assumes that there is no reliable attitude estimate, the MEKF cannot be reliably implemented in this case. This highlights why star trackers use the Wahba Problem (a nonlinear batch estimation problem) instead of using an MEKF (a sequential filter) when solving the lost-in-space problem.

3.6.7 Selecting an Optimization Routine for Use in the MOAR Algorithm

With many different optimization routines in existence, it may be difficult to select the best one for the application. The goal of the MOAR algorithm is to estimate bias quickly and accurately so it can be removed from the gyro signal and attitude can be estimated. Typically for numerical techniques there exists a trade-off between speed and accuracy, which is the case here. Local optimization techniques are typically faster and less computationally expensive than their global counterparts. However, global optimization routines cover larger search spaces and are more likely to converge to the global minimum for the cost function. By implementing a grid search, it is possible to find the best initial guesses to converge to the truth, but requires searching the entire solution space. The figures below show the performance of all 4 optimization routines: unconstrained, LM, SA, and BFGS with a grid search. The final estimates were then plotted on top of a contour plot that shows the difference in final cost function value. This was done to better represent the performance of the optimization routines, and the results can be seen in Figure 3.14.

Bias estimates for SA show good convergence, except for the z axis. For LM the convergence is slightly worse, and the estimates do not converge for BFGS, with the exception of a few cases. Because the z axis is weakly observable, the bias estimates for these optimization routines do not converge to the truth. The final values of the cost function were also analyzed to determine how accurately and quickly each algorithm minimized the problem. These results can be seen in Figure 3.15.

It is apparent for the BFGS algorithm that if the initial estimate is close to the truth, it will converge quickly with a small cost function value. When this is not the case, the final cost function value is around two orders of magnitude higher. In the case of LM, as seen previously the estimates are found when the predetermined tolerance is reached, in general, but they tend to settle to local minima more often. The final cost function values exist on a line, which is influenced by the z axis being weakly observable. For SA, other than requiring more iterations to converge, there is very little difference in final cost function value when compared to LM. The BFGS-grid estimates showed the best convergence to the lowest final

Figure 3.14: The gyro bias estimates for 200 Monte Carlo runs (blue stars) are overlaid on contours of cost function value to show the performance of the various optimization methods under consideration. The true bias is $\mathbf{b} = [2.86 \ 2.86 \ 2.86]^T$ deg/s. Only the BFGS-Grid method is able to handle the highly nonlinear and multimodal cost function in a robust manner.

cost function values with the number of iterations required similar to that of SA. Statistical data for all 4 algorithms can be seen in Table 3.3.

The data shows that the means of the bias estimates for the BFGS algorithm were significantly far away from bias truth, and the associated covariance values were on the order of 10^2 , which shows poor agreement among all estimates. In the case of LM, the means of the bias estimates were near the truth for the x and y components, however, the z component was significantly far away from truth. This disparity is reflected by the covariance of the x and y components being an order of magnitude smaller than the z components' covariance. For SA, the mean bias is very near the truth, with covariance values

Figure 3.15: Final cost function values for all four optimization techniques versus final iteration number. It is apparent that the BFGS-grid technique had the lowest average cost function values with the number of iterations required consistent with that of SA. Because of this it was chosen for use in the MOAR algorithm.

Table 3.3: Maximum, minimum, mean and standard deviation (σ) values for the bias estimates of all 4 optimization routines. The true bias for each axis was 2.8648 *deg/s*.

	BFGS (deg/s)	LM (deg/s)	SA (deg/s)	BFGS-grid (deg/s)
min x	-709.2780	-30.3242	2.7225	2.8578
min y	-601.7483	-14.4053	2.5166	2.8554
min z	-653.5963	-33.1185	-0.6805	2.8066
max x	273.7780	10.9341	3.1755	2.8717
max y	562.5083	19.0218	3.4057	2.8751
max z	2484.3	70.6239	5.1472	2.9327
mean x	7.4122	2.2095	2.8734	2.8652
mean y	6.2656	2.9860	2.8483	2.8646
mean z	-30.1182	-4.1300	2.5368	2.8655
σ_x	78.4574	2.8171	0.0555	0.0025
σ_y	63.9423	2.1122	0.0904	0.0036
σ_z	207.4240	10.5705	1.0755	0.0244

on the order of magnitude of 10^{-2} for x and y . However, it too has a higher z covariance, in this case by two orders of magnitude. The BFGS-grid mean bias estimates were almost exactly truth, showing excellent convergence with covariance values on the order of 10^{-3} for x and y . Because of the overall lower cost function values, and better convergence of the bias estimates, the BFGS-grid algorithm was chosen as the optimization routine in the MOAR algorithm.

Up until this point, bias instability has not been accounted for in the simulations, as it was assumed

to have a negligible effect on bias estimation due to the relatively short overall integration time being used. The effects that can be expected from using different quality IMUs is presented in the following section.

3.6.8 Effect of Bias Instability on Bias Estimates

Section 3.5.2 defined the bias instability property of an IMU, or how the bias value at start up may drift over time. This drift can be defined by looking at the Allan variance associated with the IMU. Different grades of IMUs have different magnitudes of bias instability, with cheaper MEMS IMUs being more susceptible than tactical and navigation grade IMUs, which show almost no effects. This difference between IMUs was simulated using bias instability 1σ standard deviations derived from data sheets [9], [69], [70], and can be seen in Figure 3.16,

Figure 3.16: Simulated performance of MEMS, tactical, and navigational grade IMUs in regard to bias instability. The true bias is $\mathbf{b} = [2.86 \ 2.86 \ 2.86]^T$ deg/s.

Looking at the figure it is evident that a MEMS grade IMU can experience a change in significant bias over time. Tactical and Navigation grade IMU show a very stable bias over the course of how long measurement collection would take for the MOAR algorithm. The MOAR device proposes using a MEMS grade IMU to determine an attitude estimate. With this in mind, a Monte Carlo simulation was performed that analyzed the potential effects of using this sensor as well as the tactical and navigation grade IMUs for attitude estimation. The 1-sigma values from before were used to generate a bias walk,

which was then added to the gyro history for each iteration in the Monte Carlo simulation. Gyro bias was estimated like before, and the results can be seen in Figure 3.17.

Figure 3.17: Bias estimate error histograms for MEMS, tactical and navigation grade IMUs

The plots from this simulation show that the MEMS grade IMU had the largest bias error, on the order of 10^0 . Errors from the tactical grade IMU showed an order of magnitude improvement, and the navigation grade IMU estimated bias the best, with errors on the order of 10^{-2} .

These results were expected, due to the higher quality sensors having a smaller 1σ value of bias instability. However, the MEMS grade IMU was chosen for implementation on the MOAR prototype due to its small size and low cost. The short expected data collection time, around 60 seconds, was also used as justification for the MEMS IMU, because the bias theoretically will not have walked significantly over this short period of time. To ensure that the bias instability simulation was accurate according to the IMU manufacturer's specs, the Allan variance plot was reproduced by simulating bias instability of a still sensor using 1σ values supplied by the datasheet. The reproduced plot may be seen in Figure 3.18.

Figure 3.18: Allan variance plot for the ADIS16405 IMU reproduced using a bias instability 1σ of 0.007 deg/s and a 1σ noise value of 0.125 deg/s

Now that the effect bias instability could have on the MOAR algorithm has been considered, another important case must be considered. The effect of using different star constellations for data collection will be discussed in the following section.

3.6.9 Effect of Using Different Constellations for Star Sightings

The preceding sections have used the stars of the Big Dipper in Ursa Minor to define LOS vectors and gyro measurements for use in the various simulations. In addition to the Big Dipper, three other constellations — Aries, Cygnus, and Orion — were considered to determine the effect that star pattern geometry had on bias estimation performance. Cygnus and Orion were chosen due to their unique geometry and large number of stars. Aries was selected because it is a simpler geometry with few stars. All of the constellations and the assumed scope trace patterns may be seen in Figure 3.19.

Using the BFGS-grid optimization routine, a simulation was performed that tested 200 bias guesses for each of the 4 constellations. Bias estimate error was then calculated and error histograms were generated for each component of each constellation. The Big Dipper, Cygnus, and Orion are all examples of constellations with well distributed star patterns, that is, they consist of a large number of stars with a unique geometry over 2 axes. As such, their error histograms were grouped together and can be seen in Figure 3.20. Each constellation's x and y components of bias error were on the order of 10^{-2} deg, and the z component of each constellation's bias error was on the order of 10^{-1} . The lower accuracy for the z -axis is expected due to the weakly observable nature of the z component of bias (see troughs along z component direction in cost function contours in Figure 3.14).

Unlike the constellations shown in Figure 3.20, Aries is an example of a nearly degenerate constellation

Figure 3.19: Star plots of all constellations used overlaid with the order of star sightings (sequence of the trace pattern).

pattern (the sequence of star observations are very nearly a straight line). When coupled with the fact that the z component of bias is weakly observable, multiple bias estimates are found to be plausible in the presence of measurement noise. The multiple solutions found when using the Aries star pattern in the Monte Carlo analysis can clearly be seen in in Figure 3.21. This result underscores the importance of selecting a well distributed star pattern and avoiding degenerate star patterns (i.e. all stars in a line or repeated measurements of the same star).

Since simulations show that the Big Dipper and Orion are well distributed star patterns, they will be targeted for use in an experimental implementation of the MOAR algorithm. In the next section, the effect that sampling stars in different orders has on the MOAR algorithm will be investigated.

Figure 3.20: Bias estimate error histograms for three well distributed star patterns: Big Dipper (top row), Cygnus (middle row), and Orion (bottom row).

Figure 3.21: Bias estimate error histograms for a nearly degenerate star pattern: Aries. Since the solution space is multi-modal, in the presence of noise there exist multiple solutions when using a nearly degenerate star pattern such as Aries.

3.6.10 Effect of Sampling Stars in Different Orders

Up to this point, all simulations use star data that has been recorded from the left-most star moving right. This consistent sequencing makes all previous simulations easy to evaluate, but in practice a different order may be used to trace star constellations. A new simulation was performed that looked at using other stars in the Big Dipper as the starting point for data collection, and also tracing out the star patterns in different directions. The results from this simulation can be seen in Figures 3.22 and

3.23.

Figure 3.22: Bias error histogram from using the Big Dipper, starting with the first star and tracing to the right (left) and starting with the last star and tracing to the left (right)

Figure 3.23: Bias error histogram from using the Big Dipper, starting with the middle star and making a right handed system, no noise (left) and noisy (right)

As expected, the error between each case was bounded by the same values, as the x , y and z bias errors were all on the order of magnitude of 10^{-2} . Intuitively this makes sense as the truth LOS vectors

from the star catalog were modified for each case to match the order of the stars as they were sighted. Assuming that the user knows which stars are being sighted, and their order the MOAR algorithm will always function as intended for bias and attitude estimation. Another consideration that could have an impact on the MOAR algorithm was the effect that varying time between measurements will have on bias estimation, which will be discussed in the next section.

3.6.11 Effect of Time Between Measurements on Solution Space

A primary concern with inputs to the MOAR algorithm is the time between star sightings. It can be expected that a trained crew member operating the device can expect to need 5-10 seconds to move from one star to the next. Several scenarios were considered where the time between star sightings was varied from 5 to 30 seconds in increments of 5, and the cost function values were analyzed. Contour plots were generated to show how the solution space changed with the time between star sightings, and they can be seen in Figure 3.24.

In each figure, it is apparent that increasing the time between star sightings reduces the size of the solution space, allowing for better convergence to the global minimum. This is especially noticeable in the z component figures, and the change between 25 and 30 seconds shows that the problem is also multi-modal. Because of this, there is an upper bound on the amount of time between star sightings that ensures that the optimization routine converges to the global solution. Realistically, however, the time between star sightings will be closer to 5-10 seconds, with stars in unique constellations easily identifiable and close to each other. As a result, there should be minimal effects from this property of the solution space when the MOAR algorithm is evaluated.

Now that all of the special concerns with the MOAR algorithm have been addressed, it is possible to effectively estimate bias and remove it from the gyro signal. An attitude estimate may then be found using the gyro corrected data. The Wahba problem was used to estimate attitude in the MOAR algorithm, and will be discussed in the next section.

3.7 Attitude Estimation

Once a bias estimate has been obtained and removed from the gyro data, it is possible to accurately determine the spacecraft's attitude. Using the bias-corrected gyro data along with LOS vectors to stars, the Wahba problem may be applied to solve for attitude. A solution to the lost-in-space attitude determination problem is sought, which is defined as the rotation between the orientation of the sensor frame at the reference time t_j (S_j) with respect to the inertial frame (I). This was done by solving the

Figure 3.24: Contour plots showing how the solution space cost function values change as time between star sightings changes

Wahba problem (first presented in [15] and summarized in Section 2.2), which attempts to minimize the following cost function,

$$\text{Min } J(\mathbf{T}_{S_j}^I) = \frac{1}{2} \sum_{i=1}^n w_i \|\tilde{\mathbf{u}}_{i_{S_j}} - \mathbf{T}_{S_j}^I \mathbf{u}_{i_I}\|^2 \quad (3.7.1)$$

where w_i is the weighting factor for the i th measurement, and the matrix $\mathbf{T}_{S_j}^I$ is the rotation matrix that describes the rotation from I to S_j . Several methods exist of solving the Wahba problem [29, 71, 72], and in this paper the Singular Value Decomposition (SVD) [29] method was chosen. The SVD approach begins by defining the attitude profile matrix, $\mathbf{B}$, as

$$\mathbf{B} = \sum_{i=1}^n w_i \tilde{\mathbf{u}}_{i_{S_j}} \mathbf{u}_{i_I}^T \quad (3.7.2)$$

where w_i is the weighting factor found from the variance of the pointing error,

$$w_i = \frac{1}{\sigma_{pt}^2} \quad (3.7.3)$$

The SVD operation is then applied to the matrix $\mathbf{B}$ such that,

$$\mathbf{B} = \mathbf{U} \mathbf{S} \mathbf{V}^T \quad (3.7.4)$$

From here, the final attitude estimate is simply

$$\mathbf{T}_{S_j}^I = \mathbf{U} \mathbf{M} \mathbf{V}^T \quad (3.7.5)$$

where

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{U})\det(\mathbf{V}) \end{bmatrix} \quad (3.7.6)$$

3.8 Simulation of the MOAR Algorithm

At this point, the groundwork has been laid for the MOAR algorithm to be simulated in its entirety. A Monte Carlo simulation was conducted that evaluated a large iterations using noisy and biased gyro data to determine an attitude estimate. By analyzing a large number of cases, a reliable probability distribution is generated that yields expected error values of the overall process. These results show the

feasibility of using the MOAR algorithm as a way to manually estimate attitude of a spacecraft. Before a Monte Carlo simulation can be performed, a true initial attitude and simulated gyro data between star measurements had to be defined.

3.8.1 Simulating Device Motion for a Sequenced Number of Stars

To begin the simulation, data were pulled from the Bright star catalog and then stored in an array. Information for stars in the desired constellation was selected by entering the corresponding star number for each desired star. The data were then used to calculate the true LOS vectors for each star in the inertial frame. To simulate motion of the device between any two star measurements, the ideal case of a single axis rotation with constant angular velocity was used. Such motion may be found by computing the vector normal to the two star LOS directions and finding the angle between these stars, as shown in Fig. 3.25.

Figure 3.25: Cross Product Between two Star Vectors

The angle between stars can be found from the inner product of the inertial LOS to star A, $\mathbf{u}_A$, and the inertial LOS to star B, $\mathbf{u}_B$,

$$\cos \theta = \frac{\mathbf{u}_A^T \mathbf{u}_B}{\|\mathbf{u}_A\| \|\mathbf{u}_B\|} \quad (3.8.1)$$

and the normal vector between the two vectors can be found by

$$\mathbf{n}_I = \frac{\mathbf{u}_A \times \mathbf{u}_B}{\|\mathbf{u}_A \times \mathbf{u}_B\|} \quad (3.8.2)$$

These two equations can then be combined to determine a reasonable angular rate in the inertial frame by defining a time step between measurements, Δt ,

$$\boldsymbol{\omega}_I = \frac{\theta}{\Delta t} \mathbf{n}_I \quad (3.8.3)$$

The inertial body rate may be converted to the sensor frame to simulate the noise-free angular velocity that would be measured by a strap-down gyroscope,

$$\boldsymbol{\omega}_{S_i} = \mathbf{T}_{S_i}^I \boldsymbol{\omega}_I \quad (3.8.4)$$

The true attitude quaternion was tracked throughout the entire simulation using the inertial angle rate data and Equations 3.5.3 and 3.5.4. Although its value is somewhat arbitrary, initial attitude had to be defined before the simulation could be performed.

3.8.2 Determining Initial Attitude

In order to determine the initial condition for integrating the true attitude solution, a frame is constructed using the first two stars. The frame is somewhat arbitrary, so long as the z-axis of the first sensor frame points at the first star. Thus, a set of orthogonal unit vectors may be created (if A is the first star and B is the second star),

$$\begin{aligned} \mathbf{x}_I &= \mathbf{n}_I \\ \mathbf{y}_I &= \frac{\mathbf{u}_A \times \mathbf{n}_I}{\|\mathbf{u}_A \times \mathbf{n}_I\|} \\ \mathbf{z}_I &= \mathbf{e}_A \end{aligned} \quad (3.8.5)$$

which may be used to construct the desired initial condition,

$$\mathbf{T}_I^{S_1} = [\mathbf{x}_I \quad \mathbf{y}_I \quad \mathbf{z}_I] \quad (3.8.6)$$

After determining the initial attitude matrix of the spacecraft, the next step before a simulation could be performed was to define the error on the LOS measurements.

3.8.3 Pointing Error Modeling

An expected major source of error in the MOAR algorithm is pointing error, an example of which may be seen in Figure 3.26. It is up to the user to properly align the scope boresight with the LOS to the star being measured. Despite the device being mounted on a tripod, there exists some degree of

pointing error, and this must be accounted for in the simulation. If it is not accounted for, the attitude estimate solved for by the MOAR algorithm will not be as reliable. In the Gemini and Apollo missions, a pointing error threshold was set at 10 *arcsec*, with experimental tests showing pointing error variances of around 7 *arcsec* [3], [4]. This same threshold of 10 *arcsec* was used in the Monte Carlo simulation for attitude estimation.

Figure 3.26: Exaggerated visualization of pointing error when targeting stars in the Big Dipper. Red lines and circles represent true LOS measurements, and green lines and circles represent pointing error.

Chapter 4

Experimental Approach and Design

In the previous chapter, the components of the MOAR algorithm were simulated and/or analyzed to provide a benchmark of what could be expected in a real-world application. The ability to estimate gyro bias independently of attitude was initially proven through these simulations. In order to further prove that these methods were sound, an experimental apparatus was constructed that was used to experimentally validate the results from the gyro estimation simulations. This chapter highlights the construction of a MOAR prototype, data acquisition methods used and error sources that affected the empirical results.

4.1 Experimental Testing Apparatus

A prototype of the MOAR device was constructed that consists of a rifle scope, a computer vision camera, a Raspberry Pi computer, an IMU and a push button attached to a momentary switch. All of these components were mounted onto a lightweight frame that was constructed out of aluminum. The frame contains two tiers separated by PVC supports; the bottom tier was dedicated to the electronics of the system, and the top tier consisted of the scope and camera. The CAD design of the prototype and actual build can be seen in Figure 4.1.

An adjustable 1-4X scope with a red illuminated donut-dot reticle was used for targeting stars. This helped the user to discern the reticle from the night sky, rather than the standard black crosshairs which would have proven difficult to use. The donut-dot reticle is a ring concentric with the center of the FOV along with a dot in the center. In addition to effectively targeting stars, this also provides the user with an ‘on-the-fly’ visual representation of how much pointing error is being introduced into the measurements.

Figure 4.1: Engineering drawing of the MOAR prototype (left) and constructed MOAR prototype (right)

Figure 4.2: Illuminated reticle of the scope used on the MOAR prototype

The necessary computing power to collect data from the IMU as well as capture images from the camera was provided by a Raspberry Pi computer. The use of the camera provided another means to verify the accuracy of experimental methods. Originally the camera used by the prototype was a simple Raspberry Pi ribbon camera with a custom designed light baffle and case, seen in Figure 4.3. However, it was determined that it could not take images bright enough to discern different stars in a low light situation. Because of this, a standard computer vision camera was chosen instead and can be seen in Figure 4.4. For the prototype, an Edmund Optics (model number EO-5012C) equipped with a 25 mm lens was chosen. By aligning the boresight of the scope with the boresight of the camera, images may be captured of the stars that the user is targeting which may then be used to compare results to star catalog data, although images from the camera were not used in this analysis.

In order to tell the system when to capture IMU data, a push button was attached to the electronics board of the prototype. Its state was constantly monitored through the GPIO pins on the Raspberry

Figure 4.3: Engineering drawing of Raspberry pi camera with light baffle and case (left) and rapid prototyped build (right)

Figure 4.4: Edmund Optics camera used on the MOAR prototype

Pi. When the button was pressed, the Raspberry Pi would read a different input state on the GPIO pin, and the data at that instance in time was recorded to a text file. The button was designed such that it could be operated away from the prototype frame. In this manner, the action of pressing the button would not produce a vibration that would corrupt sensor data.

4.1.1 Data Acquisition

In order to record data from the IMU and determine when the button to record gyro measurements was pressed, a Raspberry Pi Model B computer, seen in Figure 4.5 was used. It has 512 MB of RAM, a Broadcom BCM2835 processor, 2 USB ports, and 26 General-purpose Input/Output (GPIO) pins. A keyboard and mouse can be plugged in to the USB ports, allowing the user to enter commands via the command-line interface. The Edmund Optics camera could also be utilized through one of the USB ports.

The button, IMU, and Raspberry Pi were all interfaced through a custom printed circuit board (PCB) designed for this project, which can be seen in Figure 4.6. All data signals were transferred to

Figure 4.5: Raspberry Pi Model B used on the MOAR prototype

the GPIO bus and were distributed via copper traces. The GPIO pins were used to read IMU data and to determine when the measurement button was clicked. The Raspberry Pi runs a script from startup that records IMU data with a timestamp to a text file, and when the button is pressed the timestamp is marked with three stars (***) to denote a gyro measurement. The button was wired such that when it was pressed the state of GPIO pin 17, set to always be ‘high’ at 3.3V, would change the state to ground. This process was repeated for a total of 10 button clicks, at which point the user would have control over the Raspberry Pi in the command-line interface.

Figure 4.6: Engineering drawing of the printed circuit board used on the MOAR prototype

Powering all of the components is a 9V battery pack that is regulated down to 5V. The IMU power and output signals are transferred via the copper traces on the PCB, and a fuse is placed such that in the event of a power surge the IMU will not be affected. The Raspberry Pi may be powered using either the battery pack via the USB B-out or a wall adapter. For the experiment, the wall adapter was used to increase battery life.

4.1.2 Inertial Measurement Unit

The IMU used in the MOAR prototype was an ADIS16405 tri-axis inertial sensor with a magnetometer, and the datasheet may be seen in [9]. It was chosen because of its low cost, compact size ($23\text{ mm} \times 23\text{ mm} \times 23\text{ mm}$), serial peripheral interface (SPI), and register structure that allows for data collection at a rate of 10 Hz. The 3-axis alignment of the sensor may be seen in Figure 4.7, and it should be noted that its coordinate system differs from that of the sensor frame. The IMU was mounted onto the prototype with the its x , y and z axes aligned with the MOAR sensor z , x and y axes, respectively (again, see Figure 4.7).

Figure 4.7: Axial orientation of the ADIS16405 IMU [9] and the MOAR sensor frame

Before reliable data from the IMU could be recorded, it had to be calibrated. This process was performed using the WVU Applied Space Exploration Laboratory (ASEL) rate table, and is described in the following section.

4.2 IMU Calibration

Before data could be collected, the IMU had to be calibrated to ensure that it was outputting usable data. In order to do this, the electronics from the prototype were removed and mounted onto a bracket which was then mounted onto the ASEL rate table. The ARMS-200 rate table is a rotary motion simulator made by Aerotech Inc. that provides high-accuracy angular positions, rates, and accelerations

for the purposes of testing inertial components [73]. Since there are no gear trains or mechanical couplings in their design, positions errors due to hysteresis, windup, and backlash are eliminated. The ASEL rate table is also equipped with a patent-pending calibration rig developed by Drew Bittner and John Christian [74]. A custom built bracket allowed the MOAR device electronics to be placed inside this calibration rig, which allowed for reliable testing of each axis. The bracket and rate table set-up used for the IMU calibration can be seen in Figure 4.8.

Figure 4.8: MOAR prototype electronics mounted inside the ASEL's patent-pending calibration rig on a single axis rate table, aligned with the IMU z -axis (same as the MOAR y -axis)

Once the apparatus had been set-up, 4 different rotation rates were selected to determine if there were any inconsistencies with the IMU output. To begin, the bracket with the MOAR electronics was aligned with the desired axis. First, the sensor was turned on and allowed to sit still for at least 2 minutes. Then the rate table was ramped up, over a period of 10 seconds, to a constant angular velocity of 10 deg/s . It stayed at this angular velocity for a period of 60 seconds, and the rate table was then ramped up to 20 deg/s . The process was repeated in this manner for angular velocities of 30 and 45 deg/s , and after 60 seconds at 45 deg/s , the rate table was decelerated to 0. This process was repeated until all 3 axes had been evaluated, and the results may be seen in Figures 4.9-4.11.

This calibration procedure allowed an error to be uncovered in the original IMU code that resulted

Figure 4.9: Rate table IMU calibration results for the x gyro output, in the local IMU reference frame.

Figure 4.10: Rate table IMU calibration results for the y gyro output, in the local IMU reference frame.

in the digital data being shifted by two bits, resulting in the data being off by a factor of four. This was easily corrected and highlights the importance of performing a controlled calibration.

Figure 4.11: Rate table IMU calibration results for the z gyro output, in the local IMU reference frame.

4.3 Uncertainty Budget

In order to quantify the impact of all the noise sources on the attitude estimate produced by the MOAR prototype/algorithm, an error budget was determined. There were misalignment errors between the scope to IMU case, the IMU case to gyro, and the IMU case to the MOAR frame. The standard deviation of the IMU case to gyro misalignment was given by the vendor in the datasheet as a value of 0.5 deg . Because of machining tolerances involved in assembling the prototype, the other misalignment values were estimated to be on the order of 1.0 deg . Assuming all of these error sources are independent and uncorrelated, the total error may be found by taking the root-sum-square (RSS) of the contributing standard deviations. Thus, using these values, an overall noise deviation of 1.5 deg was determined for alignment error sources.

The errors associated with the IMU itself were ARW, residual bias, and scale factor. A 1σ value for the ARW was given by the vendor as $2 \text{ deg}/\sqrt{\text{hr}}$, and when multiplied by the square root of time between star sightings (10 sec) in hours gives an error of 0.1054 deg . For residual bias error, comparisons between static bias and estimated bias show that the values can differ by 0.002 deg/s for some cases. Using this value and 10 sec between star sightings, a worst case error arising from residual bias was estimated as 0.02 deg . The true IMU scale factor is close to (but not exactly the same as) the value provided by the vendor. Assuming a scale factor error of 5 ppm , a maximum body rate of 3 deg/s , and 10 sec between star sightings, the error between two sequential star sightings due to scale factor error

should not exceed 0.00015 *deg*. These values yield an overall noise deviation of 0.1073 *deg* from IMU errors.

Error is introduced into the system by the user in the forms of pointing and timing error. Pointing error was estimated as having a sigma value of 1 *arcmin*, based on the errors in the donut-dot reticle seen in Figure 4.2. Timing error can be quantified by looking at the rotation rate around the time of a star sighting (in this case around 0.2 *deg/s*) and multiplying it by $\frac{0.1sec}{\sqrt{12}}$ which gives a timing error of 0.0058 *deg*. These two noise sources yield an overall noise deviation of 0.0180 *deg* from user errors.

Astronomical effects also contribute to error on the star sighting measurements. Parallax, as discussed in Section 3.3.1, can introduce error because the apparent star direction changes based on the location of the observer. The case of sighting Alpha Centauri was already analyzed, and it was determined that the worst case errors due to parallax will not exceed 0.00042 *deg*. Stellar aberration is the effect caused when an observer is moving relative to the frame of the object being observed. Using equation 3.3.4 from Section 3.3.2, the β value was determined to be 1×10^{-4} for an observer standing on Earth (heliocentric velocity of 30 *km/s*). Converting this to degrees yields an error of 20.62 *arcsec* or 0.00573 *deg*. Atmospheric refraction can also introduce errors on apparent star directions, depending on its location above the horizon. In Table 1 from [75], the deviation can vary from 0 *arcmin* (when looking straight up) to 4.9 *arcmin* or 0.082 *deg* (when looking along the horizon), and the worst case value was chosen for the error budget. This yields an overall noise deviation of 0.0822 *deg* from astronomical effects. All of the error sources are compiled in Table 4.1.

Table 4.1: Worst case uncertainty budget for star sighting directions for the MOAR prototype

Noise source description	Noise value in <i>deg</i>	RSS of noise in <i>deg</i>
<i>Alignment errors</i>		<i>1.5</i>
Scope to IMU case misalignment	1.0	
IMU case to MOAR frame misalignment	1.0	
IMU case to gyro misalignment	0.5	
<i>IMU errors</i>		<i>0.1073</i>
Angle random walk	0.1054	
Residual bias error	0.02	
Scale factor error	0.00015	
<i>User errors</i>		<i>0.0180</i>
Pointing error	0.017	
Timing error	0.0058	
<i>Astronomical effects</i>		<i>0.0822</i>
Parallax	0.00042	
Stellar aberration	0.00573	
Atmospheric refraction	0.082	
<i>RSS of all noise</i>		<i>1.5062</i>

The root-sum-square (RSS) was applied to all of the error sigma values, and the overall noise covariance was determined to be 1.506 *deg*. It is apparent from this magnitude that alignment errors (at 1.5 *deg* of error) were the driving error source in the MOAR algorithm. A portion of the overall error was mitigated when the design of the MOAR device was changed to remove the button from the frame. As a result, the gyro readings will not be corrupted by the vibration from pressing the button.

Once the effects of potential error in the experiment had been considered, experimental data were collected using the procedure that will be discussed in the next section.

4.4 Experimental Procedure

Once the IMU was calibrated and all other components checked for functionality, an experiment was conducted to test the MOAR algorithm. The experimental apparatus, seen in Figure 4.12, was assembled on the roof of the Engineering Science Building on WVU's campus (Latitude 39.645512 N, Longitude 79.974065 W). This vantage point is the highest point in Morgantown, WV, and as a result eliminated a significant amount of light pollution from interfering with star sightings. The procedure that was followed for data collection can be seen below.

1. Determine a suitable testing location with an unobstructed view of the desired constellations
2. Set up prototype on the tripod and connect peripherals to the Raspberry Pi
3. Set up laptop to record experimental data after each iteration
4. Record the longitude and latitude of the testing location
5. Turn on the illuminated reticle on the scope
6. Visually identify the desired constellation
7. Supply power to the IMU and Raspberry Pi
8. Allow the prototype to sit still for 2 minutes
9. Trace out the constellation, pressing the button to mark good star LOS measurements
10. Bring the prototype back to a level position, and allow the prototype to sit still for 2 minutes
11. Shut down the Raspberry Pi, transfer data from SD card onto laptop, and verify successful data acquisition

12. Repeat Steps 6-11 for a minimum of 10 iterations per constellation
13. If desired, choose a new constellation and repeat Steps 6-11
14. Power off all components and dismantle testing apparatus

Drawing inspiration from the results of Duke [24] where the space sextant operators experienced lower errors with the scanning telescope's 20X magnification, the scope was set to its maximum magnification level at 4X for the experiment. This made stars difficult to target at first, but after a few iterations targeting stars became easier and the increased magnification made centering the desired star with the boresight easier.

Figure 4.12: Experimental setup of the MOAR prototype on March 2, 2016

This experiment was intended to collect data on both the Big Dipper and Orion constellations. However, it disappeared quickly behind the horizon after data from the Big Dipper was collected, around 0100 on March 3. Because of this, a second experiment was conducted on March 9 on the roof of the WVU Engineering Science Building, and data for the Orion constellation was collected. The data were then post-processed in Matlab, and the results are presented in the next chapter.

Chapter 5

Results

The previous chapter outlined an experimental approach and design of the MOAR prototype. This chapter will discuss results from a Monte Carlo simulation of the MOAR algorithm as well as the experimental results from the prototype.

5.1 Monte Carlo Simulation

In order to better understand the expected system performance in the presence of noisy gyro and scope LOS measurements, a Monte Carlo simulation was performed. Data was simulated for a total of 1000 cases using the error values discussed in Chapter 3. Like before, the Big Dipper was used to simulate star LOS measurements. An overall summary of the simulation parameters can be seen in Table 5.1.

Table 5.1: Summary of error values for the MOAR algorithm Monte Carlo simulation

Description	Initial Conditions / Standard Deviation
Constellation	Big Dipper
Time Between Star Sightings	$\Delta t = 10$ s
Star LOS Error	$\sigma_{pt} = 2.778 \times 10^{-3}$ deg
Gyro Noise	$\sigma_{\nu} = 0.1261$ deg/s
Initial bias guess	$[0 \ 0 \ 0]^T$
Number of Cases	1000

For each run, the attitude error was computed using the equation,

$$\delta \hat{\mathbf{T}} = \mathbf{T}_{S_j}^I (\hat{\mathbf{T}}_{S_j}^I)^{-1} \quad (5.1.1)$$

Error histograms were then produced for each axis and can be seen in Figure 5.1. For the x and y

axes, attitude error was on the order of 10^{-1} deg, showing excellent estimation of attitude given noisy and biased gyro data. Even though it was weakly observable, the z axis attitude errors were on the order of 1 deg, with a magnitude no larger than 3.14 deg of error. Statistical data on the the Monte Carlo simulation can be seen in Table 5.2

Figure 5.1: Attitude error histogram of a Monte Carlo simulation of the MOAR algorithm

Table 5.2: Maximum, minimum, mean, and standard deviation (σ) values for the bias estimates of all four optimization routines

	Min (deg)	Max (deg)	Mean (deg)	Std. dev. (deg)	Skewness	Kurtosis
attitude error x	-0.1001	0.1094	7.888×10^{-4}	0.0345	-0.0101	2.8903
attitude error y	-0.1862	0.1833	-9.426×10^{-5}	0.0576	0.1252	3.0590
attitude error z	-3.1380	2.6760	0.0514	0.9472	-0.1334	2.9606

The excellent attitude estimation capabilities of the MOAR algorithm were further highlighted by the mean errors of 7.888×10^{-4} for the x axis, -9.426×10^{-5} for the y axis and 0.0514 for the z axis. Skewness values for each axis are nearly zero, which implies that the distribution of error is nearly symmetrical. Kurtosis values for each axis are very close to 3, which coupled with the skewness values implies that

the error distribution is very nearly Gaussian. This shows that the MOAR algorithm does not introduce any bias into the system, and coupled with the low error values the conclusion can be derived that it estimates attitude very effectively, even with a noisy and biased MEMS grade IMU used in the process.

Now that the overall algorithm has been simulated, it must be verified with experimental data. The following section presents the results from field testing the MOAR prototype.

5.2 Results from MOAR Prototype Testing

After proving that the MOAR algorithm is effective at removing bias from gyro data and estimating attitude in simulation, it was necessary to prove it empirically. The following sections present the results from 2 separate experiments. The first one used the Big Dipper as the star constellation for data collection. A second experiment was conducted a week later that used the constellation Orion for data collection.

5.2.1 Test Case 1: Big Dipper

On March 2, 2016, data were collected from the roof of the Engineering Science Building on WVU's campus. The goal of this experiment was to collect data using the Big Dipper and Orion constellations as references. A total of 10 cases were recorded using the Big Dipper.

Note from the experimental procedure in Section 4.4 that each data acquisition session begins with collecting two minutes of static IMU data. This allows for a crude estimate of the steady-state bias for each experimental run. Collecting such data clearly isn't feasible on a spacecraft that may be tumbling, but it is possible in a controlled experimental setting. Although there are some complications arising from the static sensor measuring Earth's rotational rate, these issues are small.

Graphical data from this experiment may be seen below in Figure 5.2, where the top plot shows LOS vectors for each case calculated from raw gyro data, the middle plot shows LOS vectors calculated from gyro data with the observed steady-state bias removed, and the bottom plot shows LOS vectors for each case calculated after the bias has been estimated using the MOAR algorithm and removed from the gyro data.

The standard deviation for the Big Dipper bias-corrected LOS measurements in Figure 5.2 was 0.5499 *deg*. Table 5.3 shows the cost function evaluated for each of the 10 cases with no bias removed from the data, the steady-state bias removed, and the bias estimate removed from the data. The cost function values are the largest for the raw data, and there was a reduction in cost function when the steady-state bias was removed, in most cases an order of magnitude difference. As expected, the cost function values

Figure 5.2: Visualization of all ten star tracings (gray lines) of the Big Dipper plotted in the inertial frame using raw test data (top), data with steady-state bias removed (middle), and bias-corrected data (bottom). The black dots represent true LOS directions from the star catalog.

were the lowest when bias-corrected gyro data were used, with an order of magnitude improvement over the cases where the steady-state bias was removed. This affirms the graphical data seen in Figure 5.2, where the LOS vectors from the bias-corrected gyro data lie centered around the stars' true LOS vectors.

Table 5.3: Cost function values for raw data, raw data with steady-state (SS) bias removed, and MOAR bias corrected data for the Big Dipper

Cost fn raw data	Cost fn SS bias removed	Cost fn bias corrected
0.2683	0.0379	0.00045
0.1543	0.0088	0.00078
0.3262	0.0514	0.00091
0.2240	0.0762	0.00061
0.0869	0.0085	0.00041
0.1148	0.0049	0.00045
0.1397	0.0208	0.0011
0.1378	0.0416	0.00032
0.0486	0.0045	0.00029
0.0723	0.0118	0.00084

Table 5.4: Steady-state (SS) bias values derived from IMU output at the beginning of each Big Dipper star sighting compared against optimal estimate of the bias from the MOAR algorithm, in deg/s , compared to the final cost function value

SS x bias	est x bias	SS y bias	est y bias	SS z bias	est z bias
0.0027	0.0036	-0.00018	-0.0023	0.0099	0.0089
0.0036	0.0032	-0.0022	-0.0016	0.0112	0.0107
0.0023	0.0040	-0.0013	-0.0016	0.0100	0.0040
0.0020	0.0048	-0.0012	-0.0019	0.0104	0.0087
0.0023	0.0039	-0.0013	-0.0011	0.0103	0.0096
0.0026	0.0040	-0.0019	-0.0015	0.0098	0.0133
0.0021	0.0042	-0.0016	-0.0017	0.0101	0.0063
0.0027	0.0050	-0.0015	-0.0031	0.0101	0.0199
0.0020	0.0040	-0.0017	-0.0013	0.0098	0.0056
0.0020	0.0025	-0.0012	-0.0020	0.0101	0.0101

5.2.2 Test Case 2: Orion

On March 9, 2016, data were collected from the roof of the Engineering Science Building on WVU's campus. The primary goal of this second experiment was to collect data using the Orion constellation as a reference, and 10 separate star tracings were completed. As with the Big Dipper test case, static data were collected at the beginning of each data collection period to create a steady-state bias estimate.

Graphical data from this experiment may be seen below in Figure 5.3, where the top plot shows LOS vectors for each case calculated from raw gyro data, the middle plot shows LOS vectors calculated from gyro data with the observed steady-state bias removed, and the bottom plot shows LOS vectors for each

case calculated after the bias has been estimated using the MOAR algorithm and removed from the gyro data.

The standard deviation for the Orion bias-corrected LOS measurements in Figure 5.3 was 0.6846 *deg*. Table 5.5 shows the cost function evaluated for each of the 10 cases with no bias removed from the data, the steady-state bias removed, and the bias estimate removed from the data. The cost function values are the largest for the raw data, which was expected. There was a reduction in cost function when the steady-state bias was removed, in some cases an order of magnitude difference. Also as expected, the cost function values were the lowest when bias-corrected gyro data were used, with an order of magnitude improvement over the cases where the steady-state bias was removed.

Table 5.5: Cost function values for raw data, raw data with steady-state (SS) bias removed, and MOAR bias corrected data for Orion

Cost fn raw data	Cost fn SS bias removed	Cost fn bias corrected
0.2269	0.1471	0.0030
0.0395	0.0052	0.0047
0.1459	0.0216	0.0015
0.0570	0.0099	0.00091
0.0454	0.0048	0.00076
0.0409	0.0023	0.00050
0.0537	0.0147	0.0054
0.8683	0.0173	0.0042
0.7152	0.0261	0.00066
1.1742	0.0693	0.0014

Table 5.6 compares the steady-state bias observed from allowing the prototype to remain still for the beginning 2 minutes of data collection and the bias estimates determined from the MOAR algorithm. thus, the overall effectiveness of the MOAR algorithm was demonstrated again using the Orion constellation as a reference.

Table 5.6: Steady-state (SS) bias values derived from IMU output at the beginning of each Orion star sighting compared against optimal estimate of the bias from the MOAR algorithm, in *deg/s*

SS x bias	est x bias	SS y bias	est y bias	SS z bias	est z bias
0.00089	0.0036	-0.0023	-0.0018	0.0136	0.0049
0.0029	0.0026	-0.00018	0.00009	0.0176	0.0136
0.0022	0.0035	-0.00086	-0.00008	0.0109	0.0142
0.0019	0.0031	-0.00069	-0.00080	0.0106	0.0136
0.0023	0.0027	-0.00070	0.00027	0.0111	0.0122
0.0026	0.0031	-0.00081	-0.0021	0.0102	0.0041
0.0020	0.0022	-0.00017	-0.0016	0.0107	0.0327
0.0039	0.0046	-0.00052	-0.0015	0.0106	0.0122
0.0040	0.0050	-0.00038	-0.00063	0.0101	0.0108
0.0036	0.0047	-0.00085	-0.00072	0.0103	0.0089

Figure 5.3: Visualization of all ten star tracings (gray lines) of the Big Dipper plotted in the inertial frame using raw test data (top), data with steady-state bias removed (middle), and bias-corrected data (bottom). The black dots represent true LOS directions from the star catalog.

5.3 Summary of Results

This research set out to show that it is possible to accurately estimate attitude manually using biased and noisy gyro data along with scope LOS measurements. The previous 2 sections demonstrated the performance of the MOAR algorithm using 2 different constellation — the Big Dipper and Orion.

In both cases, the raw data showed how bias affected the LOS measurements, causing significant errors in attitude determination. Even removing the steady-state bias that was observed from start-up did not yield accurate LOS measurements for use in the Wahba problem. By applying the BFGS-grid optimization routine to data from both constellations, the gyro bias was effectively estimated and removed. The corrected data were then used to calculate accurate LOS measurements and the impact that this process had is immediately recognizable. Visually, the bias-corrected LOS measurements were very near to the true LOS vectors defined from the star catalog.

The reduction in cost function after removing the bias provided further proof of the MOAR algorithm’s capability. Table 5.7 shows the reduction in cost function for both constellations. For the Big Dipper, the average cost function value was reduced from 0.1573 using raw gyro data to 0.000619 using bias-corrected gyro data. Similarly, the average cost function value for Orion was decreased from 0.3367 using raw gyro data to 0.0023 using bias-corrected gyro data. These cost function reductions of 3 and 2 orders of magnitude, respectively, quantify the effectiveness of the MOAR algorithm in removing bias and estimating attitude using multiple constellations.

Table 5.7: Average cost function values of the Big Dipper and Orion tests

Constellation	Avg cost fn raw	Avg cost fn corrected
Big Dipper	0.1573	0.000619
Orion	0.3367	0.0023

It should be noted that all of the results were achieved by the author operating the prototype with it mounted on a camera tripod. No training for using the prototype to target stars was done, and this shows how easy it is to use. If this were to be implemented on a spacecraft, there would be some degree of training in its use before flight and thus the performance of the MOAR device would be even better. It should also be noted that the spotting scope had a magnification power of 4X, while the space sextant experiments of Apollo used a scanning telescope of 20X.

The research objectives outlined in the beginning of this dissertation were all achieved. The MOAR algorithm was simulated in Matlab (Objective 1) and experimentally verified through experimentation. Gyro bias was removed by comparing the interior star angles calculated from LOS measurements and star catalog data and supplying this information to a BFGS and grid search algorithm (Objective 2). In

regards to accuracy versus time required to estimate bias, the BFGS-grid algorithm provided the best results with a time requirement close to that of the global optimization routine SA. Simulation times for a single BFGS-grid case took in the neighborhood of 5 minutes on a standard desktop computer. Therefore, it was determined that this was an acceptable amount of time for the accuracy achieved (Objective 3). Finally, using the bias-corrected gyro data an accurate attitude estimate was found by the MOAR algorithm (Objective 4).

Chapter 6

Conclusions and Recommendations

6.1 Conclusions

In this work, the principles behind the MOAR algorithm, which manually determines attitude of a spacecraft using star LOS measurements and gyro data, were outlined. A computer model of the overall process was made to simulate its performance and also when certain parameters were varied. These included using different constellations for data collection, the effects of bias instability, sampling stars in different orders, and varying the time between measurements. The results showed that using constellations that are well distributed, such as the Big Dipper, Orion and Cygnus, led to smaller bias estimate errors. When a nearly degenerate constellation such as Aries was used, it led to higher errors because multiple bias estimates were found to be plausible in the presence of measurement noise. Bias instability over the course of measurement collection time in the MOAR algorithm was insignificant. However, there was a noticeable difference between the bias stability of MEMS, tactical and navigation grade IMUs. Simulations also proved that the order in which stars are observed does not matter, as long as the order is consistent between measured and true star LOS vectors. Finally, simulations showed that increasing the time between star sightings reduces the size of the solution space, allowing for better convergence to the true bias.

Significant contributions were made through the work of this dissertation to two primary areas: manual spacecraft attitude determination and bias estimation independent of attitude. Similarly to the space sextant of the Gemini and Apollo missions and the COAS of the Space Shuttle, star LOS measurements were used to calculate an attitude estimate. Normally attitude is determined autonomously by onboard star trackers and IMUs. However, if the star trackers become saturated by light or the IMU's error

grows too large, the attitude estimation algorithms will fail. The space sextant was a hand-held device and the COAS required that the shuttle be rotated to record LOS measurements. The difference with the MOAR algorithm, however, is that the rotation rates between star measurements are recorded by an IMU separate from the space vehicle, thus requiring no need to rotate the vehicle for data collection.

Typically gyro bias is determined concurrently with the vehicle's attitude using a sequential filter. In this dissertation, a method was presented that used batch estimation and an optimization technique to estimate bias independently of attitude. By minimizing the square of the magnitude of the difference between measured and interior star angles for a known constellation, the bias is able to be estimated. This novel approach requires no *a priori* knowledge of bias or attitude, allowing it to be used to solve the lost-in-space problem. It does require, however, that a well distributed star pattern be used to avoid converging to an incorrect bias that exists in the multi-modal solution space.

6.2 Recommendations for Future Work

There are several opportunities to further the research presented in this dissertation. Currently the camera that is mounted on the prototype is operated from Matlab. Future work would include writing code on the Raspberry Pi that allows images to be taken when the button is pressed. This could be used to verify the star LOS measurements, and could be made autonomous by implementing a star ID algorithm similar to a star tracker's.

A way to improve the MOAR device would be to test using a higher grade IMU on the MOAR device to yield even more accurate attitude estimates. This may not be necessary, however, as the attitude estimate provided by the MEMS grade IMU will be close enough to ensure convergence in a flight computer's sequential attitude filter. Theoretically, with a better attitude estimate provided to the onboard computer, it will require less convergence time and experience less error.

The MOAR device could also be redesigned to be more compact, as currently it is rather large and cumbersome on top of a standard camera tripod. By streamlining it into a neat package, it would be significantly easier to operate. This would allow the user to target stars more accurately and have better control over the time between measurements.

Another application for the MOAR device that warrants exploration is its use in a land-based or maritime setting. With ships relying heavily on GPS for navigation, and the use of sextants not being as popular as it was in the past, a GPS denied environment could bring a ship to a stand still since its crew would have no way to accurately navigate. In this case, the MOAR device could be used as a digital sextant that doesn't require bulky almanacs to operate in the same manner as a marine

sextant. Land-based vehicles or hikers could also make use of the MOAR device as a position check when navigating.

The final recommendation involves considering how the MOAR device would be mounted on an actual spacecraft. The rotation between the spacecraft and the mounted device would need to be determined before use in space. This has not been directly considered in this work, but an initial design concept would seek to use a mount similar to those used in helicopter door-mounted spot lights. Also, instead of making use of a device mounted computer, the spacecraft's flight computer could also be used to process star LOS and rotation data. This would allow for a smaller device as well as faster computation times.

Bibliography

- [1] Loff, S., “NASA Names New Crew Exploration Vehicle Orion,” August 2006.
- [2] Silva, R., Jorris, T., and Vallerie, E., “Experiment D005- Star Occultation Navigation,” *The Gemini Program: Physical Sciences Experiments Summary*, No. NASA TM X-58075, National Aeronautics and Space Administration, Sept 1971, pp. 51–59.
- [3] Lampkin, B. A., “Sextant sighting performance for space navigation using simulated and real celestial targets,” *NAVIGATION: Journal of the Institute of Navigation*, Vol. 12, No. 4, 1965, pp. 312–320.
- [4] Smith, D. W., “Hand-held sextant-Results from Gemini XII and flight simulator experiments.” *Journal of Spacecraft and Rockets*, Vol. 5, No. 6, 1968, pp. 655–662.
- [5] Draper, C. S., Wrigley, W., Hoag, D. G., Battin, R. H., Miller, J. E., Koso, D., and Hopkins, A., “Space navigation guidance and control,” Tech. Rep. NASA CR-75543, 1965.
- [6] Jorris, T. and Barth, A., “The USAF manned space navigation experiment on Apollo and its implications on advanced manned spacecraft,” *presented at the Institute of Navigation Meeting on Simplified Manned Guidance, Navigation and Control*, 1968.
- [7] Spall, J. C., *Introduction to Stochastic Search and Optimization*, Wiley-Interscience, New Jersey, 2003, pp. 208–213.
- [8] Kouwenhoven, T., “Celestial Reference System,” 2011, [Online; accessed February 3, 2016].
- [9] Analog Devices, *Tri-Axis Inertial Sensor with Magnetometer*, 2015, Rev. C.
- [10] Webb, D. F. and Allen, J. H., “Spacecraft and ground anomalies related to the October-November 2003 solar activity,” *Space Weather*, Vol. 2, No. 3, 2004.

- [11] Baker, D., Belian, R., Higbie, P., Klebesadel, R., and Blake, J., “Deep dielectric charging effects due to high-energy electrons in earth’s outer magnetosphere,” *Journal of Electrostatics*, Vol. 20, No. 1, 1987, pp. 3–19.
- [12] Robinson Jr, P. A., “Spacecraft environmental anomalies handbook,” Tech. rep., Jet Propulsion Laboratory, California Institute of Technology, 1989.
- [13] Merancy, N., Dehmlow, B., and Brazzel, J. P., “Holographic weapons sight as a crew optical alignment sight,” *SPIE Defense, Security, and Sensing*, International Society for Optics and Photonics, 2011, pp. 80440R–80440R.
- [14] Kalman, R. E., “A new approach to linear filtering and prediction problems,” *Journal of Fluids Engineering*, Vol. 82, No. 1, 1960, pp. 35–45.
- [15] Wahba, G., “A least squares estimate of satellite attitude,” *SIAM review*, Vol. 7, No. 3, 1965, pp. 409–409.
- [16] Agee, W. S. and Turner, R. H., “Optimal Estimation of Measurement Bias,” Tech. rep., White Sands Missile Range, 1972.
- [17] Troni, G. and Whitcomb, L. L., “Adaptive Estimation of Measurement Bias in Three-Dimensional Field Sensors with Angular Rate Sensors: Theory and Comparative Experimental Evaluation.” *Robotics: Science and Systems*, 2013.
- [18] Namvar, M. and Safaei, F., “Adaptive compensation of gyro bias in rigid-body attitude estimation using a single vector measurement,” *Automatic Control, IEEE Transactions on*, Vol. 58, No. 7, 2013, pp. 1816–1822.
- [19] Silva, R., Jorris, T. R., and Vallerie, E. M., “The Air Force space navigation experiment on Gemini (DOD/NASA Gemini Experiment D-9, Gemini IV and VII Flights),” Tech. Rep. AFAL-TR-66-289, 1966.
- [20] Carroll, J. and Lillestrand, R., “Dynamic space navigation employing star occultation technique,” Dec 1967, US Patent 3,360,638.
- [21] Shufeldt, H. H., “Precision Celestial Navigation Experiments,” *Journal of Navigation*, Vol. 15, No. 03, 1962, pp. 301–324.
- [22] Murtagh, T., Price, C., and Smith, H., “Analysis of Gemini 7-star sightings utilizing a space sextant in Gemini 6.” *Journal of Spacecraft and Rockets*, Vol. 4, No. 5, 1967, pp. 567–572.

- [23] Hoag, D. G., “The history of Apollo on-board guidance, navigation, and control,” *In: The Eagle has returned; Proceedings of the Dedication Conference of the International Space Hall of Fame, Alamogordo, N. Mex., October 5-9, 1976.(A77-26926 11-12) San Diego, Calif., American Astronautical Society; Univelt, Inc., 1976, p. 270-300.*, Vol. 1, 1976, pp. 270–300.
- [24] Duke Jr, C. M., *Human performance during a simulated Apollo mid-course navigation sighting*, Ph.D. thesis, Massachusetts Institute of Technology, 1964.
- [25] Kachmar, P. M. and Wood, L. J., “Space navigation applications,” *NAVIGATION: Journal of the Institute of Navigation*, Vol. 42, No. 1, 1995, pp. 186–234.
- [26] Brennan, P. and Johnson, I., “Manually-aided on-board Apollo Cislunar navigation,” *The Role of Man in Navigation*, Institute of Navigation, 1970.
- [27] Walsh, R. C. and Ferguson Jr, J., “The United States Air Force manual space navigation experiment on Skylab/DOD/NASA Skylab experiment T-002,” *American Institute of Aeronautics and Astronautics (AIAA) and American Astronomical Society (AAS), Astrodynamics Conference*, 1976.
- [28] Sarafin, M., “Guidance, Navigation, and Control Section 18. Crew Optical Alignment Sight,” *National Aeronautics and Space Administration (NASA) Shuttle Operations Handbook, JSC-18863*, 1998.
- [29] Markley, F. L., “Attitude determination using vector observations and the singular value decomposition,” *The Journal of the Astronautical Sciences*, Vol. 36, No. 3, 1988, pp. 245–258.
- [30] Venkataraman, P., *Applied optimization with MATLAB programming*, John Wiley & Sons, 2009.
- [31] Papadimitriou, C. H. and Steiglitz, K., *Combinatorial optimization: algorithms and complexity*, Courier Corporation, 1998.
- [32] Nocedal, J. and Wright, S., *Numerical optimization*, Springer Science & Business Media, 2006.
- [33] Mangasarian, O. L., “Pseudo-convex functions,” *Journal of the Society for Industrial & Applied Mathematics, Series A: Control*, Vol. 3, No. 2, 1965, pp. 281–290.
- [34] Mathews, J. H. and Fink, K. D., *Numerical methods using MATLAB*, Vol. 31, Prentice hall Upper Saddle River, NJ, 1999.
- [35] Bonnans, J.-F., Gilbert, J. C., Lemaréchal, C., and Sagastizábal, C. A., *Numerical optimization: theoretical and practical aspects*, Springer Science & Business Media, 2006.

- [36] Davidon, W. C., "Variable Metric Method for Minimization," *SIAM Journal on Optimization*, Vol. 1, No. 1, 1991, pp. 1–17.
- [37] Fletcher, R. and Powell, M. J., "A rapidly convergent descent method for minimization," *Computer journal*, Vol. 6, No. 2, 1963, pp. 163–168.
- [38] Vanderplaats, G. N., *Numerical optimization techniques for engineering design: with applications*, Vol. 1, McGraw-Hill New York, 1984.
- [39] Broyden, C. G., "The Convergence of a Class of Double-rank Minimization Algorithms," *IMA Journal of Applied Mathematics*, Vol. 6, No. 1, 1970, pp. 76–90.
- [40] Fletcher, R., "A new approach to variable metric algorithms," *The Computer Journal*, Vol. 13, No. 3, 1970, pp. 317–322.
- [41] Goldfarb, D., "A family of variable-metric methods derived by variational means," *Mathematics of Computation*, Vol. 24, No. 109, 1970, pp. 23–26.
- [42] Shanno, D. F., "Conditioning of quasi-Newton methods for function minimization," *Mathematics of Computation*, Vol. 24, No. 111, 1970, pp. 647–656.
- [43] Levenberg, K., "A method for the solution of certain non-linear problems in least squares," *Quarterly of Applied Mathematics*, 1944, pp. 164–168.
- [44] Marquardt, D. W., "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial & Applied Mathematics*, Vol. 11, No. 2, 1963, pp. 431–441.
- [45] Moré, J. J., "The Levenberg-Marquardt algorithm: implementation and theory," *Numerical Analysis*, Springer, 1978, pp. 105–116.
- [46] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E., "Equation of state calculations by fast computing machines," *The journal of chemical physics*, Vol. 21, No. 6, 1953, pp. 1087–1092.
- [47] Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., et al., "Optimization by simulated annealing," *science*, Vol. 220, No. 4598, 1983, pp. 671–680.
- [48] Ingber, L., "Very fast simulated re-annealing," *Mathematical and computer modelling*, Vol. 12, No. 8, 1989, pp. 967–973.

- [49] Szu, H. and Hartley, R., “Fast simulated annealing,” *Physics letters A*, Vol. 122, No. 3, 1987, pp. 157–162.
- [50] LaValle, S. M., Branicky, M. S., and Lindemann, S. R., “On the relationship between classical grid search and probabilistic roadmaps,” *The International Journal of Robotics Research*, Vol. 23, No. 7-8, 2004, pp. 673–692.
- [51] Kavraki, L. E., Švestka, P., Latombe, J.-C., and Overmars, M. H., “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *Robotics and Automation, IEEE Transactions on*, Vol. 12, No. 4, 1996, pp. 566–580.
- [52] Huang, Q., Mao, J., and Liu, Y., “An improved grid search algorithm of SVR parameters optimization,” *Communication Technology (ICCT), 2012 IEEE 14th International Conference on*, IEEE, 2012, pp. 1022–1026.
- [53] Hesterman, J. Y., Caucci, L., Kupinski, M. A., Barrett, H. H., and Furenlid, L. R., “Maximum-likelihood estimation with a contracting-grid search algorithm,” *Nuclear Science, IEEE Transactions on*, Vol. 57, No. 3, 2010, pp. 1077–1084.
- [54] Ureten, S., Yongaçoglu, A., and Petriu, E., “Iterative grid search for RSS-based emitter localization,” *Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European*, IEEE, 2014, pp. 1507–1511.
- [55] Kalman, R. E. and Bucy, R. S., “New results in linear filtering and prediction theory,” *Journal of Fluids Engineering*, Vol. 83, No. 1, 1961, pp. 95–108.
- [56] Markley, F. L., “Attitude error representations for Kalman filtering,” *Journal of guidance, control, and dynamics*, Vol. 26, No. 2, 2003, pp. 311–317.
- [57] Crassidis, J. L. and Junkins, J. L., *Optimal estimation of dynamic systems*, CRC press, 2011.
- [58] Markley, F. L., “Multiplicative vs. additive filtering for spacecraft attitude determination,” *Proceedings of the 6th Conference on Dynamics and Control of Systems and Structures in Space (DCSSS)*, Vol. 22, 2004.
- [59] Ma, C., Arias, E., Eubanks, T., Fey, A., Gontier, A.-M., Jacobs, C., Sovers, O., Archinal, B., and Charlot, P., “The international celestial reference frame as realized by very long baseline interferometry,” *The Astronomical Journal*, Vol. 116, No. 1, 1998, pp. 516.
- [60] Einstein, A., *Relativity: The special and the general theory*, Princeton University Press, 1916.

- [61] Seidelmann, P. K., *Explanatory supplement to the astronomical almanac*, University Science Books, 2005.
- [62] Ochsenbein, F., Bauer, P., and Marcout, J., “The Vizier database of astronomical catalogues,” *A&AS*, Vol. 143, Apr 2000, pp. 23–32.
- [63] Shuster, M. D., “Stellar aberration and parallax: a tutorial,” *Journal of Astronautical Sciences*, Vol. 51, No. 4, 2003, pp. 477–494.
- [64] Christian, J. A., *Optical Navigation for a Spacecraft in a Planetary System*, Ph.D. thesis, The University of Texas at Austin, 2010.
- [65] Farrenkopf, R., “Analytic steady-state accuracy solutions for two common spacecraft attitude estimators,” *Journal of Guidance, Control, and Dynamics*, Vol. 1, No. 4, 1978, pp. 282–284.
- [66] Yuan, R., Song, N., and Jin, J., “Autonomous estimation of angle random walk of fiber optic gyro in attitude determination system of satellite,” *Measurement*, Vol. 45, No. 6, 2012, pp. 1362–1366.
- [67] Stockwell, W., “Angle random walk,” *Crossbow Technology*. [Online]. <http://www.xbow.com/pdf/AngleRandomWalkAppNote.pdf>, 2003.
- [68] Allan, D. W., “Should the classical variance be used as a basic measure in standards metrology?” *Instrumentation and Measurement, IEEE Transactions on*, Vol. 1001, No. 2, 1987, pp. 646–654.
- [69] Systron Donner Inertial, *MEMS Quartz Tactical Inertial Measurement Unit*, 2016, Rev. K.
- [70] Honeywell, *HG9900 NAVIGATION-GRADE INERTIAL MEASUREMENT UNIT (IMU)*, 2009.
- [71] Keat, J. E., “Analysis Of Least-Squares Attitude Determination Routine DOAOP,” Tech. Rep. CSC/TM-77/6034, Computer Sciences Corp, 1977.
- [72] Shuster, M. D. and Oh, S., “Three-axis attitude determination from vector observations,” *Journal of Guidance, Control, and Dynamics*, Vol. 4, No. 1, 1981, pp. 70–77.
- [73] Aerotech Inc., *ARMS Series Rotary Motion Simulators*, 2016.
- [74] Bittner, D. E., “Advances in MEMS IMU Cluster Technology for Small Satellite Applications,” 2015.
- [75] Thomas, M. E. and Joseph, R. I., “Astronomical refraction,” *Johns Hopkins apl technical digest*, Vol. 17, No. 3, 1996, pp. 279.

Appendices

Appendix A- IAU Constellation Charts

Figure 6.1: IAU constellation charts for Aries (top left), Ursa Major (top right), Orion (bottom left), and Cygnus (bottom right)

Appendix B- MOAR Data

Collection Notes

Table 6.1: Start, stop and elapsed time data from MOAR prototype experiment 1: Big Dipper, March 2, 2016. Run numbers with asterisks denote bad measurement data due to user error.

Run No.	Start Time	End Time	Elapsed Time
1	2316	2321	5:33
2	2335	2340	5:23
3	2343	2349	5:35
4	2352	2357	5:12
5	0001	0005	4:51
6	0019	0024	4:57
7	0026	0031	5:06
8*	0033	0039	6:07
9	0042	0047	5:00
10	0051	0056	5:10
11	0058	0103	5:08

Table 6.2: Start, stop and elapsed time data from MOAR prototype experiment 2: Orion, March 9, 2016. Run numbers with asterisks denote bad measurement data due to user error.

Run No.	Start Time	End Time	Elapsed Time
1	1954	1959	5:33
2	2003	2008	5:00
3*	2010	2015	5:00
4*	2018	2023	4:45
5	2027	2032	5:00
6	2034	2039	5:00
7	2043	2048	5:00
8*	2050	2058	7:45
9	2100	2105	5:10
10	2108	2113	5:00

Table 6.3: Start, stop and elapsed time data from MOAR prototype experiment 2: Orion, March 11, 2016. Run numbers with asterisks denote bad measurement data due to user error.

Run No.	Start Time	End Time	Elapsed Time
1	2152	2157	5:33
2*	2205	2211	5:33
3	2215	2220	5:33
4	2227	2233	6:00

Appendix C- MOAR Prototype

Code

```
#include <bcm2835.h>
#include <iostream>
#include <cmath>
#include <bitset>
#include <fstream>
#include <ctime>
#include <iomanip>
#include <string>
#include <sstream>
#include <unistd.h>
#include <cstdio>
#include <cstdlib>
#include <cerrno>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include "IMU.h"

const timespec stall={0,9};
const timespec burst_stall={0,1024};

using namespace std;

class GPIOClass
{
public:
    GPIOClass();
    GPIOClass(std::string gnum);

    ~GPIOClass();
    int setdir_gpio(std::string dir);
    int setval_gpio(std::string val);
    int getval_gpio(std::string& val);

    std::string get_gpionum();
```

```

int export_gpio();
int unexport_gpio();
private:

std::string gpionum;
int valuefd;
int directionfd;
int exportfd;
int unexportfd;
};

int main(int argc, char **argv){

IMU::Device my_imu; //Create instance of Device class
ofstream datastorage ("/tmp/data.txt"); //opening a file to store the data
datastorage << std::setprecision(16);

string inputstate;
GPIOClass* gpio17 = new GPIOClass("17");

//gpio17->export_gpio();

cout << " GPIO pins exported" << endl;

gpio17->setdir_gpio("in");

cout << " Set GPIO pin directions" << endl;

if (datastorage.is_open() && my_imu.is_usable()){

int ii=1;

while(ii<11)
{
usleep(100000);
gpio17->getval_gpio(inputstate);
my_imu.read();

////////////////////DATA STORAGE////////////////////////////////////
datastorage << my_imu.get_time() << ","
<< my_imu.get_supply_voltage() << ","
<< my_imu.get_xgyro() << ","
<< my_imu.get_ygyro() << ","
<< my_imu.get_zgyro() << ","
<< my_imu.get_xaccl() << ","
<< my_imu.get_yaccl() << ","
<< my_imu.get_zaccl() << ","
<< my_imu.get_temperature() << ","
<< my_imu.get_roll() << ","
<< my_imu.get_pitch() << ","
<< my_imu.get_aux_voltage() << std::endl;

// cout << "Current input pin state is " << inputstate <<endl;
//cout << my_imu.get_xaccl() <<"," << my_imu.get_yaccl()<<","<< my_imu.get_zaccl()<<endl;

```

```

// Checks GPIO pin to see if the button's pressed
if(inputstate == "0")
{
cout<<"input pin state is \"Pressed \".\n Will check input pin state again in 20ms "<<endl;
usleep(20000);
cout << "Checking again ....." << endl;
gpio17->getval_gpio(inputstate);

// Double checks the pin to make sure it's actually pressed
if(inputstate == "0")
{
cout << "input pin state is definitely \"Pressed\". Recording data" <<endl;
cout << " Waiting until pin is unpressed....." << endl;

while (inputstate == "0"){
gpio17->getval_gpio(inputstate);
};

cout << "pin is unpressed" << endl;

// Instead of all of the stuff below:
//my_imu.read();

//////////////////////DATA STORAGE//////////////////////////////////////
datastorage << my_imu.get_time() << "***" << ","
<< my_imu.get_supply_voltage() << ","
<< my_imu.get_xgyro() << ","
<< my_imu.get_ygyro() << ","
<< my_imu.get_zgyro() << ","
<< my_imu.get_xaccl() << ","
<< my_imu.get_yaccl() << ","
<< my_imu.get_zaccl() << ","
<< my_imu.get_temperature() << ","
<< my_imu.get_roll() << ","
<< my_imu.get_pitch() << ","
<< my_imu.get_aux_voltage() << std::endl;

// Increase star count
ii++;

}
}
}

datastorage.close(); //closing the data storage file

} else {
cerr << "The file tried to open. Please try again." << endl;
}

return 0;
}

```