NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

DESIGN AND SIMULATION OF A NANO-SATELLITE ATTITUDE DETERMINATION SYSTEM

by

Jason Tuthill

December 2009

Thesis Co-Advisors:	Marcello Romano Hyunwook Woo
Second Reader:	James Newman

Approved for public release; distribution unlimited

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2009	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE Design and Simulation of a Nano-Satellite Attitude Determination System	**5. FUNDING NUMBERS**
6. AUTHOR(S) Jason Tuthill	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE**

13. ABSTRACT

Earth imaging satellites have typically been large systems with highly accurate and expensive sensors. With the recent push for Operationally Responsive Space, Earth imagining has become potentially achievable with small and relatively inexpensive satellites. This has led to the research currently underway to develop very small, low-cost imaging satellites that can produce useful Operational-level and Tactical-level imagery products. This thesis contributes to that effort by developing a detailed design for the attitude determination system for a tactically useful earth-imaging nano-satellite. Tactical Imaging Nano-sat Yielding Small-Cost Operations and Persistent Earth-coverage (TINYSCOPE) is an ongoing investigation at NPS, concerning using a nano-satellite, based on the CubeSat standard, to achieve Earth imaging from LEO orbit.

A detailed design of the attitude determination system includes sensor selection and characterization, as well as high fidelity simulation via MATLAB®$^{®}$/Simulink®$^{®}$. The attitude determination system is based on an Extended Kalman Filter using multiple sensor types and data rates. The sensors include a star tracker, Sun Sensor, Gyroscope, and Magnetometer.

14. SUBJECT TERMS Kalman Filter, Attitude Determination, CubeSat, Nano-Satellite, IMU, Magnetometer, Star Tracker, Gyroscope	**15. NUMBER OF PAGES** 149
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

THIS PAGE INTENTIONALLY LEFT BLANK

DESIGN AND SIMULATION OF A NANO-SATELLITE ATTITUDE DETERMINATION SYSTEM

Jason D. Tuthill
Lieutenant, United States Navy
B.S., Rensselear Polytechnic Institute, 2001

Submitted in partial fulfillment of the
requirements for the degrees of

ASTRONAUTICAL ENGINEER
and
MASTER OF SCIENCE IN ASTRONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2009

Author:	Jason D. Tuthill
Approved by:	Marcello Romano Co-Thesis Advisor
	Hyunwook Woo Co-Thesis Advisor
	James Newman Second Reader
	Knox Milsaps Chairman, Department of Mechanical and Astronautical Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Earth imaging satellites have typically been large systems with highly accurate and expensive sensors. With the recent push for Operationally Responsive Space, Earth imagining has become potentially achievable with small and relatively inexpensive satellites. This has led to the research currently underway to develop very small, low-cost imaging satellites that can produce useful Operational-level and Tactical-level imagery products. This thesis contributes to that effort by developing a detailed design for the attitude determination system for a tactically useful earth-imaging nano-satellite. Tactical Imaging Nano-sat Yielding Small-Cost Operations and Persistent Earth-coverage (TINYSCOPE) is an ongoing investigation at NPS, concerning using a nano-satellite, based on the CubeSat standard, to achieve Earth imaging from LEO orbit.

A detailed design of the attitude determination system includes sensor selection and characterization, as well as high fidelity simulation via MATLAB®/Simulink®. The attitude determination system is based on an Extended Kalman Filter using multiple sensor types and data rates. The sensors include a star tracker, sun sensor, gyroscope, and magnetometer.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ABBREVIATIONS AND ACRONYMS

1U	One 10 x 10 x 10 cm CubeSat
3U	Three 1U CubeSats connected in series
5U	Five 1U CubeSat connected in series
6U	Two 3U CubeSats connected in parallel
ADACS	Attitude Determination and Control System
ADS	Attitude Determination System
ARW	Angular Random Walk
API	Application Programming Interface
arcsec	Arc Second: 1arcsec = 15°/3600
Cal Poly	California Polytechnic University
COTS	Commercial Of The Shelf
EKF	Extended Kalman Filter
ESPA	Evolved Expendable Launch Vehicle Secondary Payload Adapter
IEEE	Institute of Electrical and Electronics Engineers
IMI	IntelliTech Microsystems, Inc
IMU	Inertial Measurement Unit
LSS	Lab for Space Systems
MEMS	Microelectromechanical systems
mNm	miliNewton·meter
mNms	miliNewton·meter·second
NASA	National Aeronautics and Space Administration
NPS	Naval Postgraduate School
NPSCuL	NPS CubeSat Launcher
P-POD	Poly Picosatellite Orbital Deployer
PSD	Power Spectral Density
ODS	Orbit Determination System
RMS	Root Mean Square
ROM	Rough Order of Magnitude

RRW	Rate Random Walk
RSS	Root Sum Square
SEP	Spherical Error Probability
SFL	Space Flight Laboratory
SPL	Single Pico-Satellite Launcher
SSDL	Space Systems Development Laboratory
SSP	Subsolar Point
T-POD	Tokyo Pico-satellite Orbit Deployer
UAV	Unmanned Aerial Vehicle
UTIAS	University of Toronto Institute for Aerospace Studies
X-POD	eXperimental Push Out Deployer

ACKNOWLEDGMENTS

The author would like to acknowledge the financial support of the National Reconnaissance Office.

The author would like to thank the following individuals for their invaluable assistance in the completion of this thesis:

My beautiful and understanding wife, Laura, who kept things under control at home while I spent hours upon hours at school trying to finish this research.

My wonderful children, Logan and Lucas, who’s endless supply of joy and energy refreshed my spirits every morning and evening.

Dr. Hyunwook Woo for his expertise, guidance, and not only finding most of my errors, but helping me fix them.

Dr. Marcello Romano for his expertise, guidance, and support throughout this project.

Mr. Mark Looney of Analog Devices and Mr. Doug Sinclair of Sinclair Interplanetary for amazing customer service and expert advice.

Mr. Paul Oppenheimer, Maj. Chad Melone, and the rest of the NACL crew for sharing knowledge and wisdom on all-things-satellites.

The Faculty and Staff at NPS for investing so much into our education and development as Space Cadre Officers.

THIS PAGE INTENTIONALLY LEFT BLANK

I. BACKGROUND

A. BRIEF HISTORY

Since the very early days of satellite imagery, resolution and pointing accuracy have been key parameters of the satellite design. Corona, America's first Earth Imaging satellite [1], had a resolution of 8–10m. This was eventually improved to 2–4m resolution through improved cameras and lower altitudes. As subsequent generations of military Earth imaging satellites were being developed, the resolution continued to improve, which in turn lead to the need for higher accuracy pointing. The new satellites used ever-increasingly complex and expensive attitude determination systems (ADS) to provide the ability to meet these new requirements.

Today, commercial imagery satellites are going through the same trend in resolution. They have improved in the past decade, from relatively low resolution at about 5m to about 0.6m [2]. At the same time, these commercial satellites are getting smaller and keeping costs to a minimum. These two facts present new challenges to the ADS. The traditional sensors are too big and expensive to be practical for the new satellites. New sensors have been shrinking due to smaller electronics; however, this miniaturization is not keeping pace with the rapidly increasing demands on the ADS.

The latest Earth imagers take the trend of shrinking satellites to a new level. They are pico-satellites based on a new standard call CubeSats. They push the bounds of small size and low cost. Developed to provide cheap access to space and used primarily by universities, these satellites are measured in the tens of centimeters, weigh between one and ten kilograms, and usually cost in the tens of thousands of dollars. CubeSats may become a viable Earth imaging platform to complement large and very high-resolution existing assets. Therefore, the ADS is becoming a much more important part of the design. High accuracy attitude determination is still in its infancy for CubeSats. This thesis will attempt to develop an ADS that meets the needs of this new class of satellite.

B. CUBESAT STANDARD

In 1999, California Polytechnic University (Cal Poly), San Luis Obispo, and the Space Systems Development Laboratory (SSDL) at Stanford University collaborated to develop a standard for a pico-satellite design [3]. The goal was to enable rapid development and launch of satellites in the most cost effective manner. The target audience was universities and research institutes and there are currently over 100 universities, high schools, private firms, and government agencies developing CubeSats. The concept would enable groups to test new and innovative hardware and software on actual satellites instead of relying on simulations. The standardized size and mass would enable a standardized launcher to be developed that would be suitable for secondary or tertiary launch opportunities. It also enables the development of large numbers of experienced Astronautical Engineers. A student can now, in principle, with sufficient available funding and professional support, design, build, test, and launch a satellite during his undergraduate or graduate school period.

The CubeSat standard also allows for expansion from a single cube (1U). Multiple cubes can be attached together to form larger CubeSats like the 3U (3 x 1 cubes). The number of cubes attached together is really only limited by the launching mechanism. Launchers that can handle 5U (5 x 1 cubes) and 6U (2 x 3 cubes) are under development. These larger CubeSats can accommodate payloads that are more complex. The flexibility of this standard is also reflected in the fact that it is an open standard. Not just the original creators, but also the entire community of users continuously review the standard. An open standard can evolve to meet the needs of the community. The latest revision to the CubeSat mechanical requirements (Rev 12) are summarized in Table 1 and Figure 1. Another important design consideration for flexibility and utility was the electronics that would make up the working parts of the CubeSat. The sizing of the CubeSats allows for commercial electronics in the PC/104 form factor and commercial solar cells to be used. This opens a large array of previously existing systems to be used in the CubeSats. On the same point, it allows for easy development of new electronic boards and instruments using this form factor.

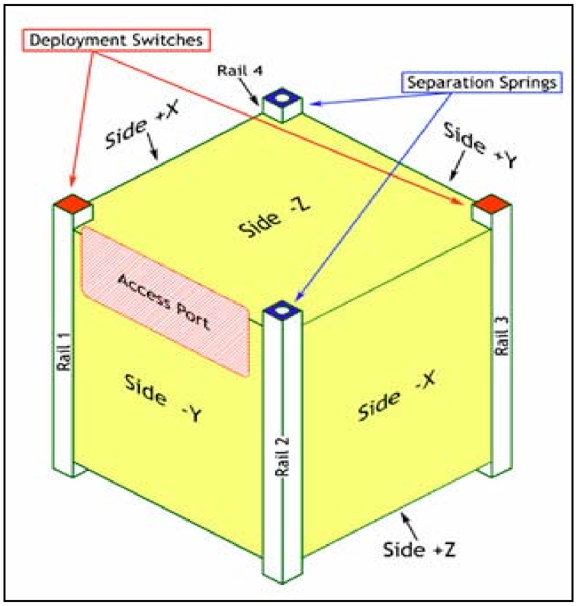


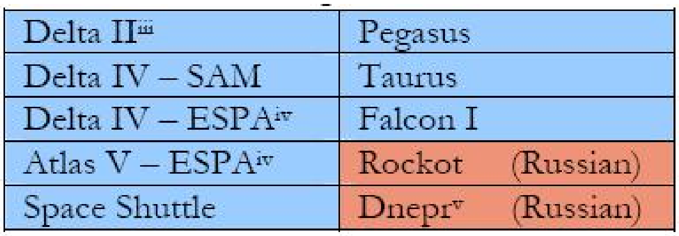
Figure 1. 1U CubeSat Side Definition. From [4].

#	Requirement	Value	Unit
1	X and Y dimensions	100±0.1	mm
2	Z dimensions (1U)	113.5±0.3	mm
3	Z dimensions (3U)	340.5±0.3	mm
4	Maximum component protrusion from X or Y side	6.5	mm
5	Mass (1U)	1.33	kg
6	Mass (3U)	4.0	kg
7	Center of gravity and geometric center difference	20	mm

Table 1. CubeSat Mechanical Specifications. After [4].

C. CUBESAT DEPLOYMENT

Cal Poly has also created a deployment standard Poly Picosatellite Orbital Deployer (P-POD). This is a box structure that houses the CubeSat during launch (see Figure 2). It provides a standard attachment point to the launch vehicle and a deployment mechanism for the CubeSat. It protects the CubeSat, launch vehicle and the primary payload. This is particularly important because the owners of the primary payload can be very conservative and want guarantees that their very expensive satellite is not damaged by the secondary payload. To ensure this, the P-POD has been tested to very high standards and proven through numerous launches that it does this job very well. It has also proven to be very flexible. It is compatible with a number of launch vehicles (see Table 2), and any CubeSat can be launched from it. This means that if the manifested CubeSat cannot launch for some reason, another one can replace it very quickly and easily. This is because any CubeSat made to the CubeSat Standard fits in a P-POD and the launch vehicle only cares about the P-POD. This combination of fitting many launch vehicles and flexible payload manifesting enables much easier access to space than ever before. Currently the P-POD can hold up to three 1U or one 3U CubeSats. There is currently research underway to develop a 5U P-POD and a 2 x 3 or 6U P-POD at Cal Poly.



Delta II[iii]	Pegasus
Delta IV – SAM	Taurus
Delta IV – ESPA[iv]	Falcon I
Atlas V – ESPA[iv]	Rockot (Russian)
Space Shuttle	Dnepr[v] (Russian)

Table 2. Launch vehicles compatible with CubeSat P-PODs. From [3].

Figure 2. CubeSat P-POD Unit. From [3].

The Naval Postgraduate School (NPS) is currently working with Cal Poly to meet the need for higher capacity CubeSat launches. The collaboration is developing a high capacity CubeSat launcher that will be designed to attach to the Evolved Expendable Launch Vehicle Secondary Payload Adapter (ESPA). The NPS CubeSat Launcher (NPSCuL) will be a collection of 10 P-PODs (see Figure 3) with coordinating electronics for CubeSat deployment. NPSCuL will be able to deploy a combination of 1U, 3U, 5U, or 6U CubeSats.

Cal Poly is not the only developer of CubeSat launchers; several other CubeSat launchers have been developed and launched [5]. Tokyo Institute of Technology's Lab for Space Systems (LSS) has developed the Tokyo Pico-satellite Orbit Deployer (T-POD). Germany's Astrofein has developed the Single Pico-Satellite Launcher (SPL). Both of these are 1U deployers. The most used

Cubesat launcher, other than the P-POD, is the eXperimental Push Out Deployer (X-POD) developed by University of Toronto Institute for Aerospace Studies/Space Flight Laboratory (UTIAS/SFL). At least six X-PODs have launched to date. They come in a variety of configurations including 1U, 3U, and the X-POD DUO holds a satellite 20 x 20 x 40 cm. NASA has even started developing a CubeSat launcher to help further develop their CubeSat program. They are specifically designing a 6U or “Six Pack” deployer.

Figure 3. NPSCuL Model. From [6].

D. SURVEY OF CUBESAT ATTITUDE DETERMINATION SYSTEMS

The majority of the CubeSat mission that have launched so far have some sort of attitude sensing (see Table 3). However, these sensors are usually low accuracy sensors like magnetometers or sun sensors. This is due to several factors. First, most of the missions flown so far do not have active attitude control, which leads to the assumption that attitude determination is also of little use. Second, most of the CubeSats with active 3-axis control typically use only magnetorquers. These provide a simple low-cost control but are also low accuracy; again, no need for high accuracy attitude determination. Finally, the high accuracy ADS is much more complex and expensive to implement. A very high accuracy sensor (<100 arcsec), like a star tracker, has yet to be flown. There are several currently in development specifically for CubeSats, but only one currently commercially available that can fit in a CubeSat.

Organization	CubeSat	Launch	ADS
Tokyo Institute of Technology	CUTE-I	2003	MEMS Gyro, 2-Axis Accelerometer, Sun Sensor
University of Tokyo	XI-IV	2003	None
University of Toronto	CanX-1	2003	Horizon sensor & Star tracker, Magnetometer, GPS
Technical University of Denmark	DTUsat	2003	MEMS sun sensor, Magnetometer
Alborg University	AAU Cubesat	2003	Sun sensor, Magnetometer
Stanford University and Quakesat LLC	Quake Sat	2003	Magnetometer
Norwegian U of Science and Technology	NCube2	2005	Magnetometer
University of Würzburg	UWE-1	2005	N/A
University of Tokyo	XI-V	2005	Camera
Tokyo Institute of Technology	CUTE 1.7 + APD	2006	Gyro, Magnetometer , Sun sensor, Earth sensor
University of Illionis	ION	2006	Sun sensors, magnetometer
University of Arizona	Sacred	2006	N/A
University of Kansas	KUTEsat Pathfinde	2006	Magnetometer, sun sensors
Cornell University (New York state)	ICE Cube	2006	Magnetometer, GPS receiver
University of Arizona	RINCON 1	2006	N/A
Nihon University	SEEDS	2006	Gyro sensor, and magnetometer
Hankuk Aviation University	HAUSAT	2006	Sun sensor, gps receiver
Norwegian U of Science and Technology	Ncube 1	2006	Magnetometer
Montana State University	MEROPE	2006	N/A
Aerospace Corporation	AeroCube-1	2006	N/A
California Polytechnic Institute	CP2	2006	magnetometer
California Polytechnic Institute	CP1	2006	Sun sensor
Cornell University (New York state)	ICE Cube 2	2006	Magnetometer, GPS receiver
University of Hawaii	Mea Huaka	2006	N/A
Center for Robotic Exploration and Space Technolo	Gene Sat-1	2006	N/A
California Polytechnic Institute	CP4	2007	magnetometer
Aerospace Corporation	AeroCube-2	2007	N/A
The Boeing Company	CSTB-1	2007	Sun sensors, magnetometer
Tethers Unlimited	MAST	2007	GPS receiver
California Polytechnic Institute	CP3	2007	Magnetometers
University of Louisiana	CAPE-1	2007	N/A
University of Sergio Arboleda	Libertad-1	2007	GPS receiver
University of Toronto	CanX-2	2008	Sun sensors, magnetometer
Tokyo Institute of Technology	II	2008	Gyro, Magnetometer , Sun sensor, Earth sensor
Delft University of Technology	Delfi-C3	2008	Sun sensor
Alborg University	AAUsat-2	2008	N/A
Fachhochschule Aachen	Compass One	2008	Magnetometer, sunsensors
Nihon University	SEEDS 2	2008	Gyro sensor, magnetometer
Hawk Institute of Space Sciences	Hawksat-1	2009	N/A
Santa Clara University, NASA Ames	Pharmasat-1	2009	N/A
California Polytechnic State University	Polysat CP6	2009	Magnetometers
Texas A&M University	Aggiesat-2	2009	N/A
University of Texas at Austin	BEVO 1	2009	N/A

Table 3. CubeSat Attitude Determination Methods. After [5].

Another dimension to this is the nearly complete lack of a pre-packaged ADS. Until August of 2009, there was only one ADS available on the market. It was the Pumpkin IMI-100 ADACS (Attitude Determination and Control System). Since then, another system has been introduced. It is the SFL CubeSat Compact 3-Axis Attitude Actuator and Sensor Pack with Sinclair Interplanetary (SFL ADCS). Both of these will fit the CubeSat Standard and provide attitude determination and control of about one degree. They differ in included equipment and performance.

1. Pumpkin IMI ADCS

Pumpkin, Inc. sells this system as a kit [7]. It includes the IMI-100 ADACS developed by IntelliTech Microsystems, Inc. (IMI), a magnetometer, and custom software. A separate sun sensor can also be supplied, but the standard setup uses only designated solar cells on the CubeSat. This sun sensor setup turns the entire CubeSat into a 3-axis sun sensor. The software simply takes the calibrated analog signal from the three sides of the CubeSat that can see the sun and uses that as the sun vector. The magnetometer is a PNI Micromag3 with a 32 nT resolution. It must be mounted outside of the IMI-100 to get uncorrupted data. The IMI-100 control equipment consists of three reaction wheels and three torque coils both orthogonally arranged. The reaction wheels can provide 1.11 mNms of momentum and 0.635 mNm of torque. A more powerful ADACS is also available—the IMI-200, which provides 2.23 mNm of torque and 10.8 mNms of momentum [8]. These two systems are compared in Table 4. A microprocessor is integrated into the package to perform all the attitude calculations and control of the actuators. A pointing accuracy of 1° is advertized with the supplied sensors. Higher accuracy is also claimed possible using a Ring Laser Gyro (RLG) and a star tracker.

	IMI-100	**IMI-200**	**Units**
Momentum Storage	1.1	10.8	mNms
Max Torque	0.635	2.23	mNm
Dimensions	10x10x79	7.6x7.6x7	Cm
Mass	907	915	Gm
Power (max)	4.32	4.32	W
Suitable SC Mass	9	18	kg

Table 4. Specifications of IMI-100 and IMI-200. After [7][8].

Figure 4. 2U CubeSat with IMI-100 ADACS (left) and IMI-100 ADACS alone (right). From [7].

2. SFL ADCS

This system is also an actuator and sensor package designed for CubeSats [9]. This ADCS has three reaction wheels and three magnetorquers, arranged orthogonally, like the IMI. It also comes with a magnetometer and six sun sensors. It, however, does not come with a computer or software. The reaction wheels are made by Sinclair Interplanetary [10] and are comparable to the IMI-200 with 10 mNms of momentum and 1 mNm of torque. There are four sun sensors integrated into the main box with two additional sensors provided to be placed on the CubeSat elsewhere. The magnetometer comes on the deployable boom to prevent magnetic interference from the satellite. Actuator and sensor specifications are listed in Table 5. This ADCS also has an advertized pointing accuracy of 1–2° rms.

SFL ADCS	**Reaction Wheel**	**Value**	**Unit**	**Magnetorquer**	**Value**	**Unit**
	Momentum Storage	10	mNms	Dipole	~0.1	Am^2
	Max Torque	1	mNm			
	Dimensions	5x5x3	cm	Dimensions	8x8x0.4	cm
	Mass	120	g	Mass	100	g
	Power (Max)	0.7	W	Power (Max)	170	W
	Sun Sensor	Value	Unit	Magnetometer	Value	Unit
	Resolution	<0.5	°	Dynamic range	±100	μT
	Accuracy	<1.5	°	Resolution	15, 30	nT
	Dimensions	3x3x1	cm	Dimensions	2x4x2	cm
	Mass	<6	g	Mass	<30	g
	Power	<0.175	W	Power	<0.025	W

Table 5. SFL/Sinclair ADCS Specifications. After [9][10].

A note for both of these systems is they only provide good attitude determination in sunlight. So, if the CubeSat goes into eclipse, which nearly all will, the attitude solution is either degraded, propagated with a model, or simply set to a nominal value. This issue is being addressed now by different institutions by incorporating other sensors, i.e., Star Tracker or Horizon Sensor, and new algorithms.

E. TINYSCOPE REQUIREMENTS

The requirements for TINYSCOPE were originally derived in [11], further refined in [12]. These requirements are the basis of this ADS design. However, the research here will benefit any nano or pico-satellite that requires either high accuracy attitude knowledge or data fusion of several different attitude sensors. The primary objective of this thesis is to design and implement a Kalman Filter that uses several separately developed techniques to combine different types of sensor data in a computationally efficient manor and produce an attitude solution that is significantly more accurate than any one of the sensors alone. The secondary objectives of this thesis are to implement realistic simulations of the various sensors used by the Kalman Filter and to generally increase the fidelity of the current Simulink® simulation model.

THIS PAGE INTENTIONALLY LEFT BLANK

II. SELECTION OF ATTITUDE DETERMINATION HARDWARE

A. PURPOSE

The development of high accuracy attitude determination for CubeSat class satellites is still new. The accuracy that is demanded by TINYSCOPE is greater than any other satellite of comparable size that has flown thus far. This means that better sensors must be developed or sensors that existed must be used in such a way to obtain better results.

B. BASIC ADS DESIGN

The ADS developed for this thesis uses commercially available (COTS) sensors. Only minor modification will need to be done to select parts. It will also be robust using five different types of sensors to ensure continuous solutions and allowances for sensor failure. The key part of this is fusing the data in such a way that the attitude knowledge is more accurate than any of the individual sensors. The ADS also cannot be dependant on one sensor to provide acceptable accuracy. Having a number of different types of sensors and using an Extended Kalman filter is the selected solution for the ADS.

The sensor types selected are mostly inertially referenced. The one exception to this is the magnetometer, which is referenced to the Earth’s magnetic field. The other sensors will include gyroscopes, sun sensors, and star trackers. A GPS receiver will also be incorporated. However, it will not be used for the attitude determination; it will be part of the orbit determination system (ODS). This system will not be developed in great detail for this thesis.

C. COMPONENT SELECTION

An extensive search for sensors that would provide high accuracy attitude knowledge for a CubeSat class satellite was conducted over the six-month period of September 2008 to March 2009. There were four primary criteria used to determine what sensor would be most useful.

1. Size. Given the CubeSat standard sensors need to be able to fit inside less than 1U.
2. Power. A CubeSat can only produce approximately twenty watts of power total.
3. Weight. Due to the CubeSat standard.
4. Performance. This was the deciding factor for many of the sensors.

Fortunately, the first three criteria tend to go hand in hand, simplifying the selection process. Other considerations were cost, the material properties, and electrical interfaces of the sensor, but these were lower order factors.

1. Inertial Measurement Unit

The Inertial Measurement Unit (IMU) is one of the key components to the ADS. It is actually a combination of two different sensors. The IMU has gyroscopes that measure the angular rate of the spacecraft and accelerometers that measure acceleration. It will provide attitude information from this sensor suite in a near continuous fashion. The EKF will specifically use the gyroscope information to estimate the attitude when the other sensors are not providing information. This will be discussed in greater detail in Chapter VI. The accelerometers are not currently being used, but could be integrated into the attitude estimation or the orbit estimation later. This type of IMU is used often on unmanned aerial vehicles (UAV) which do not generally require highly accurate sensors for flight control. Therefore, there is no real driver for the industry to produce very accurate, low noise gyroscopes and accelerometers. The gyroscopes performance was graded primarily on two characteristics: the resolution and the noise. The resolution had to be smaller than the required angular rate knowledge. The noise only had to be as small as possible because it would be compensated with filtering.

From the beginning, commercial MEMS sensors were the primary focus of the selection process. In general, they are the only gyroscope technology that meets the first three criteria for this sensor. The problem is they generally have more noise than the other types of gyroscopes do. The IMU that was eventually selected was the Analog Devices ADIS16405 (see Figure 5). A summary of the IMU specifications can be seen in Table 6 and the full specification sheet is in the Appendix. This IMU has several very useful features that could be exploited in the future. It has three dynamic ranges that are set be software. This means that during different modes of satellite operation the dynamic range could be changed to suite the expected angular rate range. The ADIS16405 also has low-pass filtering. A Bartlett window is provided by two cascaded averaging filters. The number of taps for each averaging stage can be set through software. Bias compensation can be done with either an automatic routine on the IMU or through manual settings. Both are done through software settings.

Gyroscope	Value	Unit
Dynamic Range	±300, ±150, ±75	°/s
Scale Factor	0.05, 0.025, 0.0125	°/s
Initial Bias Error	±3	°
Bias Stability	0.007	°/s
Angular Random Walk	2	°/√hr
3 dB Bandwidth	330	Hz
Dimensions	23 x 23 x 23	mm
Mass	16	g
Power	0.35	W

Table 6. Analog Devices ADIS 16405 Gyroscope Characteristics. After [13].

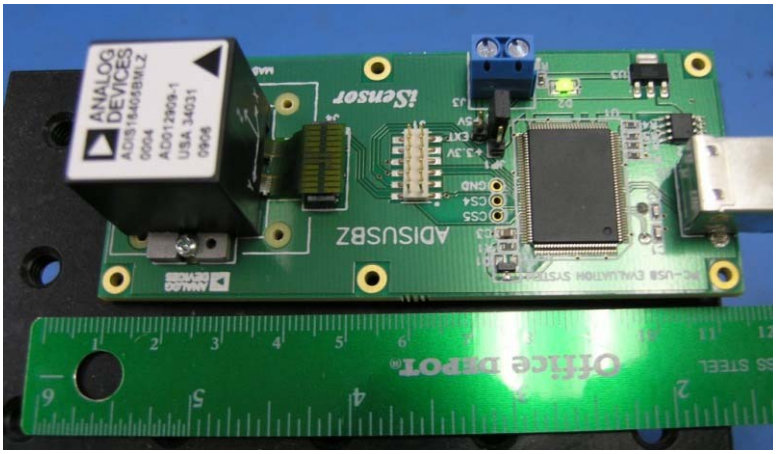


Figure 5. ADIS16405 with Evaluation Board.

2. Magnetometer

This sensor provides coarse attitude information to the ADS. Magnetometers used for attitude determination alone cannot achieve highly accurate results. This is because they need a reference model to compare the measurement to produce a usable vector. This magnetic field difference vector can then be used to determine the spacecraft's attitude with respect to the Earth. The magnetometers have become very accurate and small; however, the reference models are not very accurate. The best models use a twelve pole equivalent model, but even these cannot completely describe the Earth's complex and time changing magnetic field. One degree rms is usually considered to be about as accurate as a magnetometer ADS can get. It has also been shown that information provided by the magnetometer can be used to determine the satellite orbit [14]. This will not be pursued in this thesis, but could be added to the ODS later.

The magnetometer was graded on noise and resolution like the gyroscope. The resolution had to be small enough that the changes in the Earth's magnetic field could be detected in low Earth orbit. The noise had to be low enough that it did not drown out the measurements, but it too is filtered. The magnetometer selected is actually integrated into the ADIS16405 IMU. It provided comparable performance to other magnetometers considered at no extra cost. The characteristics of the magnetometer are summarized in Table 7.

Magnetometer	Value	Unit
Dynamic Range	±3.5	gauss
Scale factor	0.5	mgauss
Initial Bias Error	±4	mgauss
Output Noise	1.25	mgauss rms
Dimensions	23 x 23 x 23	mm
Mass	16	g
Power	0.35	W

Table 7. Analog Devices ADIS16405 Magnetometer Characteristics. After [13].

3. Sun Sensor

The sun sensors provide an important measurement for sun pointing mode and coarse measurements during the normal mode of operation. Multiple sun sensors will be needed because each one can only provide two axes of information. They are typically arranged to provide full (or nearly full) spherical coverage.

Selection for the sun sensor was primarily determined by accuracy and size. Most sun sensors that have higher accuracy are much too large for a CubeSat. Other sensors that were either very small or integrated into solar cells were too inaccurate. The sun sensor selected is the Sinclair Interplanetary SS-411 Sun Sensor. The characteristics are summarized in Table 8 and the full specification sheet is in the Appendix.

Sun Sensor Parameter	Value	Unit
Accuracy	±0.1	°
Field of View	±70	°
Bandwidth	5	Hz
Dimensions	34 x 32 x 21	mm
Mass	34	g
Power	0.075	W

Table 8. Sinclair Interplanetary SS-411 Characteristics. After [15]

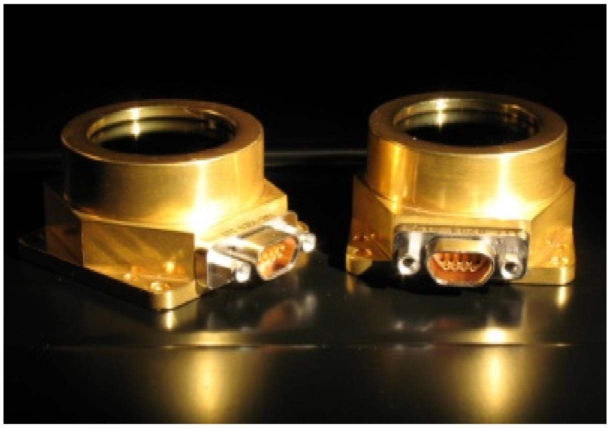


Figure 6. Sinclair Interplanetary SS-411. From [15].

4. Star Tracker

The star tracker provides the high accuracy, or fine pointing, information for the ADS. These sensors are essentially cameras that take pictures of stars. They use the star light intensity and the relative positions to determine what stars are in the picture. Once this is established, through one of many possible algorithms, the sensor attitude can be determined. Each star tracker generally provides three axes of information, so only one unit is necessary. The accuracy of a typical star tracker on a large spacecraft is a few arcseconds (arcsec). This accuracy necessarily degrades as the sensor gets smaller and the optics less refined.

The criteria for selection of the star tracker were primarily size and power. This is because very few of these sensors can even fit in a CubeSat. Ultimately, though, accuracy and availability were the determining factors. A rough estimate of the required accuracy of less that 100 arcsec[1] was made during the selection process. The selection based on these criteria became easy because at the time only one available product existed, the Commtech AeroAstro Miniature Star Tracer (MST) [16] (see Figure 7). Unfortunately, even though this is a relatively inexpensive star tracker, it is still way outside a university budget, at roughly $250,000.00[2]. Therefore, this sensor will not be purchased, but its characteristics are summarized in Table 9, and the full specification sheet is in the Appendix.

Figure 7. Commtech AeroAstro MST. From [16].

---

[1] 100 arcsec = 0.028°

[2] $250k is a ROM quote, but is (currently) accurate and includes the baffle design.

Star Tracker	Value	Unit
Accuracy	±70 (3-axes 3σ)	arcsec
Sensitivity	Up to 4th magnitude stars	
Dimensions	5.4x5.4x7.6	cm
Mass	425	g
Power	< 2	W

Table 9. AeroAstro MST Characteristics. After [16].

5. GPS

The GPS unit will provide position and timing information for the satellite. The position information will be the primary part of the ODS. The timing will be used for the spacecraft computing capabilities in general. Small size and low power were again the main factors in reducing the number of possible candidates. Both terrestrial and space qualified units were considered to help enlarge the pool of possibilities. The terrestrial units need minor modifications to work in space, so this was not of great concern. The GPS receiver selected is the NovAtel OEMV-1G-L1-A (seen in Figure 8). It is a terrestrial GPS that has been modified and flown in space previously. It was chosen because of its relatively low power requirements and good performance characteristics (see Table 10). Of interest, the GPS calculations only use approximately 40% of the processing power. The version with the Application Programming Interface (API) was ordered to allow user defined programs to run on the GPS processor. The excess processing power could be used to run the ODS. The full specification sheet is in the Appendix.

GPS	Value	Unit
L1 SEP[3]	1.8	m
Time Accuracy	20	ns rms
Dimensions	46x71x13	mm
Mass	21.5	g
Power	1	W

Table 10. NovAtel OEMV-1G Characteristics. After [17].

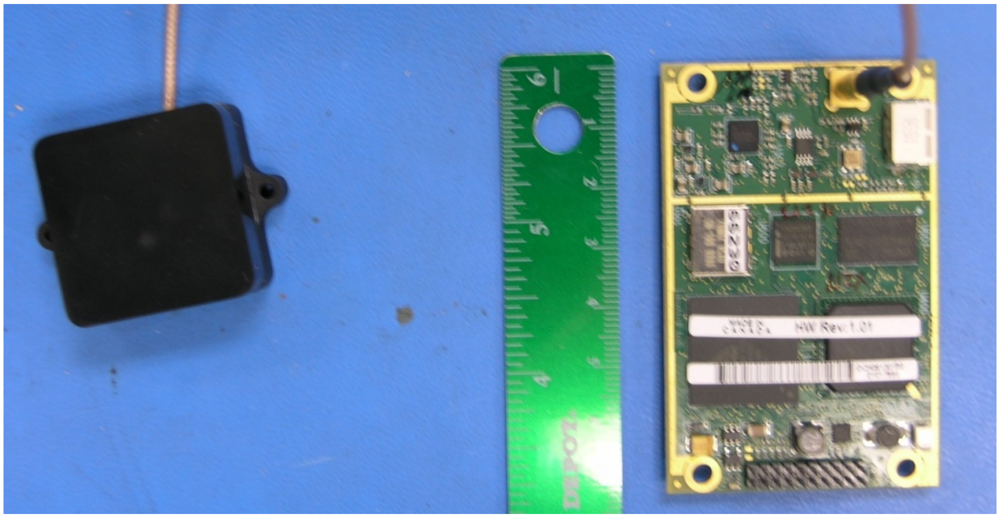


Figure 8. NovAtel OEMV-1G-L1 with L1\L2 Antenna.

---

[3] Spherical Error Probability (SEP), means the true position will be within a sphere with the given radius fifty percent of the time.

THIS PAGE INTENTIONALLY LEFT BLANK

III. EXPERIMENTAL CHARACTERIZATION OF SELECTED SENSORS

There are many ways to determine the noise sources of an instrument. Two of the most popular are the Power Spectral Density (PSD) and the Allan variance. This thesis will use the Allan variance because it is easily computed and there is existing data from the manufacturer available.

A. ALLAN VARIANCE

The Allan variance, an accepted IEEE standard for gyroscope specifications, is a time domain analysis technique that can be used to find the characteristics of the noise processes in an instrument. The Allan variance technique uses a clustering method. It divides the data into clusters of specific length and averages the data in each cluster. It then computes the variance of each successive cluster average to form the Allan variance. Each noise source has a different correlation time. By choosing the correct correlation time or cluster length, the desired noise source variance can be calculated. The Allan variance is typically plotted as the root Allan variance, $\sigma_A = \sqrt{\sigma_A^2}$ on a log-log scale. The different noise sources can be discriminated by examining the varying slopes of the root Allan variance. A more in depth discussion of Allan variance can be found in annex C of [18] or [19]. The general equations for the Allan variance forming *K* clusters from *N* data points taken at f_s samples per second with *M* points per cluster are as follows:

$$\bar{\omega}_k(M) = \frac{1}{M}\sum_{i=1}^{M}\omega_{(k-1)M+i} \qquad k = 1,2,\ldots K \qquad K = \frac{N}{M}$$

$$\underbrace{\omega_1,\omega_2,\ldots,\omega_M}_{k=1},\underbrace{\omega_{M+1},\ldots,\omega_{2M}}_{k=2},\ldots,\underbrace{\omega_{N-M},\ldots,\omega_N}_{k=K} \tag{1.1}$$

$$\sigma_A^2(\tau_M) \equiv \frac{1}{2}\left\langle\left(\bar{\omega}_{k+1}(M) - \bar{\omega}_k(M)\right)^2\right\rangle \cong \frac{1}{2(K-1)}\sum_{k=1}^{K-1}\left(\bar{\omega}_{k+1}(M) - \bar{\omega}_k(M)\right)^2 \tag{1.2}$$

Where ‹› is the ensemble average and $\tau_M = M / f_s$ is the correlation time.

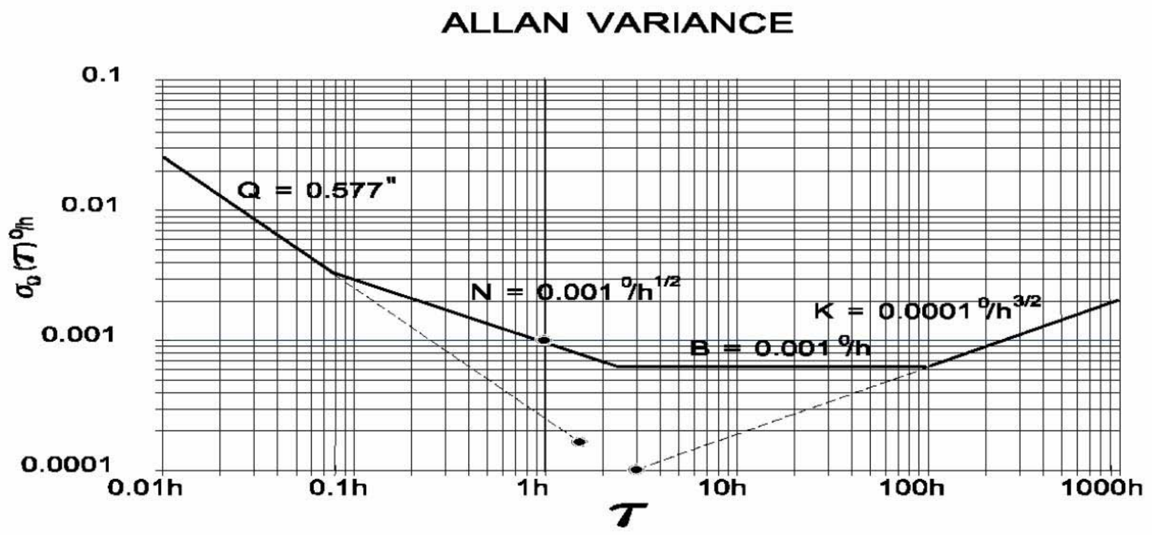


Figure 9. Piecewise Representation of Hypothetical Gyro in Allan Variance. From [18].

The following equations have been determined to calculate the variance for the Angular Random Walk (ARW) and the Rate Random Walk (RRW) noise sources in a gyro and can be found in [18]. The coefficients can be directly determined from the root Allan variance in Figure 9.

$$\sigma_{ARW}^2(\tau) = \frac{N^2}{\tau} \tag{1.3}$$

$$\sigma_{RRW}^2(\tau) = \frac{K^2 \tau}{3} \tag{1.4}$$

where *N* is the variance coefficient at $\tau = 1$ along the +1/2 slope and *K* is the variance coefficient at $\tau = 3$ along the -1/2 slope. These two noise sources are the primary noise sources being modeled in Chapter IV.

The Allan variance algorithm used to estimate noise parameters for the gyros was from MATLAB®Central, an open exchange of files for MATLAB® users. The specific code, allan v1.71, developed by M. A. Hopcroft [20] uses the basic method of Allan variance calculation developed in [19] and has been validated using the example data form [21].

B. CHARACTERIZING

1. Gyro Noise

A test was performed on the IMU to measure the noise characteristics of the gyroscopes. The test procedure from section 12.11 of [22] was referenced to setup and perform each test. The IMU was sampled with evaluation board using a USB interface and software provided by Analog Devices, Inc. The evaluation setup was performed per [23]. All tests were conducted at ambient temperature; no environmental temperature control equipment was used. To account for this the temperature was allowed to stabilize and was also recorded for each test. The magnetic field was also not controlled due to lack of equipment to do so. All testing was done in a static condition. Several tests were conducted using different sample rates, number of samples, and filtering settings (number of averaging taps).

- Gyro Data 1: Data were taken from only the Z axis gyro. It was intended to verify the method used here by comparing the calculated graph to the graph provided on the specification sheet. The following parameters were used:
    - Sample rate: 115 Hz (10 ms delay)
    - Sample points: 2,000,100 (~ 4.8 hrs of data)
    - Gyro Range: 300 °/sec
    - Filter taps: 1 (minimum low-pass filtration)
- Gyro Data 2: Data were taken from only the Z axis gyro. It will calculate the noise coefficients for the expected operation mode with no filtering. The following parameters were used:
    - Sample rate: 111 Hz (10 ms delay)
    - Sample points 2,000,100 (~4.8 hrs of data)
    - Gyro Range: 75 °/sec
    - Filter taps: 16 (minimum low-pass filtration)
- Gyro Data 3: Data were taken from only the Z axis. It will calculate the noise coefficients for the expected operation mode. The following parameters were used:
    - Sample rate: 111 Hz (10 ms delay)

- Sample points 2,000,100 (~5 hrs of data)
- Gyro Range: 75 °/sec
- Filter taps: 64 (maximum low-pass filtration)

2. Magnetometer Noise

The magnetometer noise was tested in the same way as the gyro. The same conditions exist for these tests as the gyro noise tests. The only difference is the data was collected from the magnetometers and not the gyros. Several tests were conducted using different sample rates and number of samples.

- Magnetometer Data 1: Data were taken from only the Z axis magnetometer. It was intended to find the variance in the measurements. The following parameters were used:
    - Sample rate: 115 Hz (10 ms delay)
    - Sample points: 2,000,100 (~4.8 hrs of Data)
- Magnetometer Data 2: Data were taken from only the Z axis magnetometer. It was intended to find the variance in the measurements and determine the effect of sample rate. The following parameters were used:
    - Sample rate: 210 Hz (5 ms delay)
    - Sample points: 2,000,100 (~2.65 hrs of Data)

3. Sun Sensor Noise

The sun sensor noise is previously characterized by Sinclair Interplanetary. Two tests are performed to ensure proper calibration of the sensor before it is shipped. The first measures the noise across the entire field of view. The test consists of a collimated light source moved across the sensors full field of view. Readings are taken to determine the accuracy of the sensor at each position of the light source. The dominating noise source is a spatial distortion due to variations in the optics and sensor electronics. This noise is calibrated down to a 0.1 °rms. The second test measures the time varying noise. The collimated light source is held in one position and data is taken to determine the noise level. This noise is not significant when compared to the 0.1 °rms spatial noise.

4. Star Tracker Noise

The star tracker was not purchased and so cannot be characterized. For all modeling purposes, the star tracker noise is derived from the published accuracy.

C. RESULTS & COMPARISON

This section will show the results of the above data runs. The calculated Allan variance plot and the statistics plots will be compared and explained.

1. Allan Variance for Gyro

The first objective for this testing was to verify that the calculations was providing accurate data. To do this a data set was taken with parameters to closely mimic the same conditions of the manufacturer testing, this was Gyro Data 1. The calculated Allan variance was then compared to the Allan variance provided on the specification sheet. This comparison can be see in Figure 10 and Figure 11.

It is clear that the calculated Allan variance is very similar to the manufacturer's Allan variance. The noise coefficients from the calculated Allan variance plot were also checked against the specification sheet. The ARW coefficient is 0.028 °/√sec while the specification is 0.033 °/√sec. This is reasonably close to say that the difference is due to small variations in each unit. The bias stability coefficient was also found for comparison. The coefficient from the calculations is 0.0052 °/sec while the specification is 0.007 °/sec. This again seems within reasonably manufacturing variation. It is, therefore, reasonable to validate the Allan variance calculations and assume further calculations will produce reasonable results.

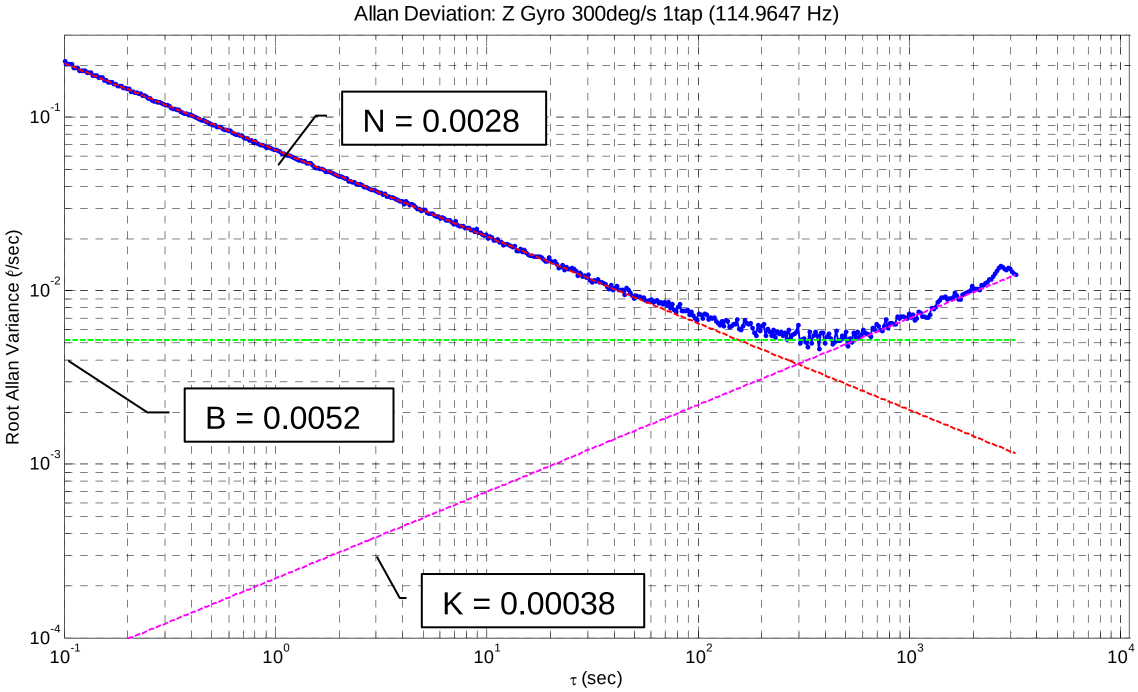


Figure 10. Calculated Root Allan Variance from Gyro Data 1.

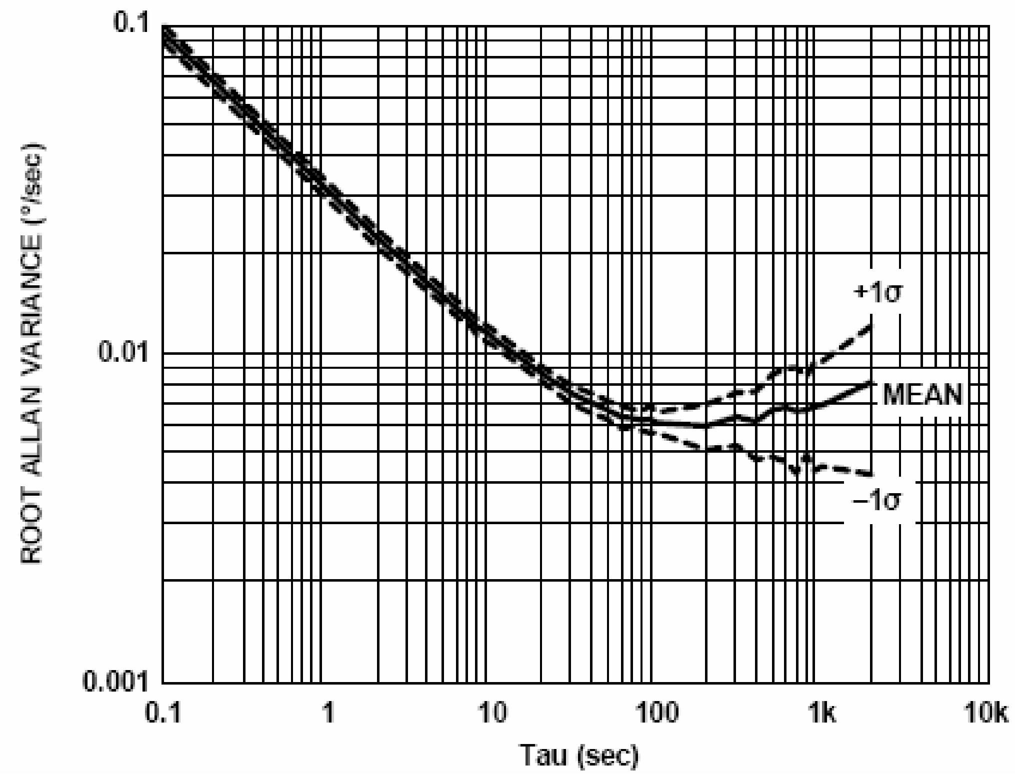


Figure 11. ADIS16405 Root Allan Variance. From [13].

The next objective is to calculate the noise sources under parameters that more closely resemble the actual operating mode. Gyro Data 2 is a data set taken under such conditions with minimal filtering applied. This will help determine the effect of the low-pass filtering. Figure 12 shows the root Allan variance plot and the noise coefficients.

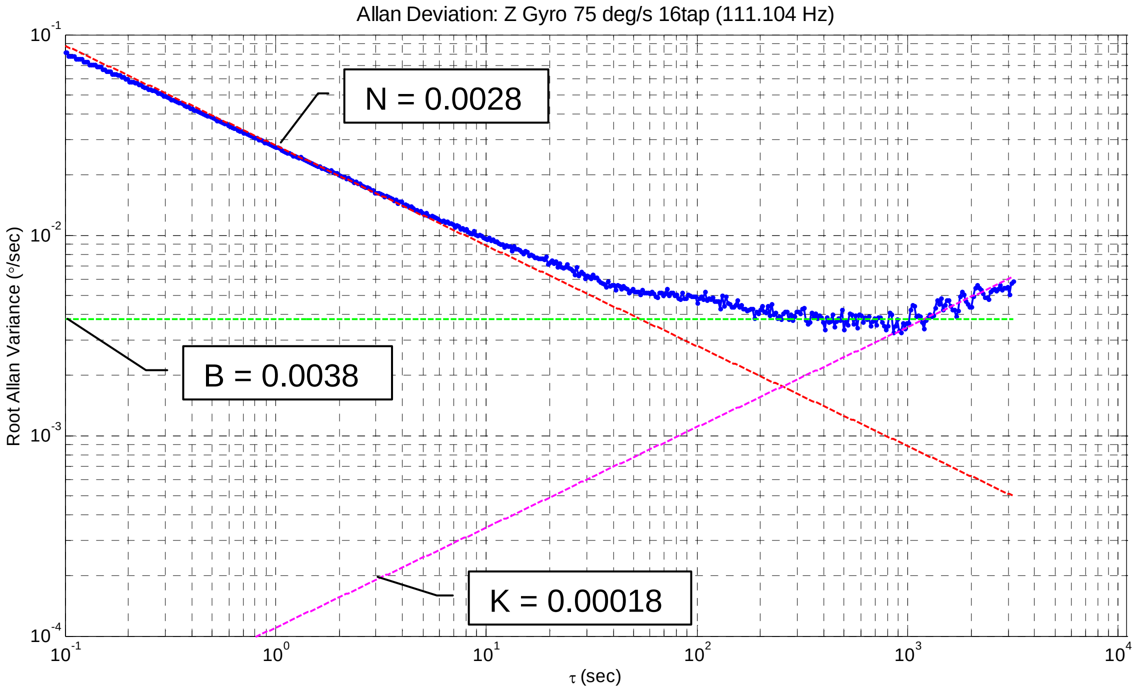


Figure 12. Calculated Root Allan Variance form Gyro Data 2.

The root Allan variance clearly maintains the overall shape seen in the previous figures. It can also be seen that, except for ARW, the noise has been reduced. The minimal low-pass filtering may be the cause of this. The noise reduction does not appear dramatic, but could be significant for improved performance of the EKF.

The last set of data, Gyro Data 3, was taken under operating parameters with full low-pass filtering applied. It is expected that the filtering will further reduce the noise and improve performance. Smaller values for all the noise coefficients will reflect this noise reduction. The root Allan variance plot can be seen in Figure 13.

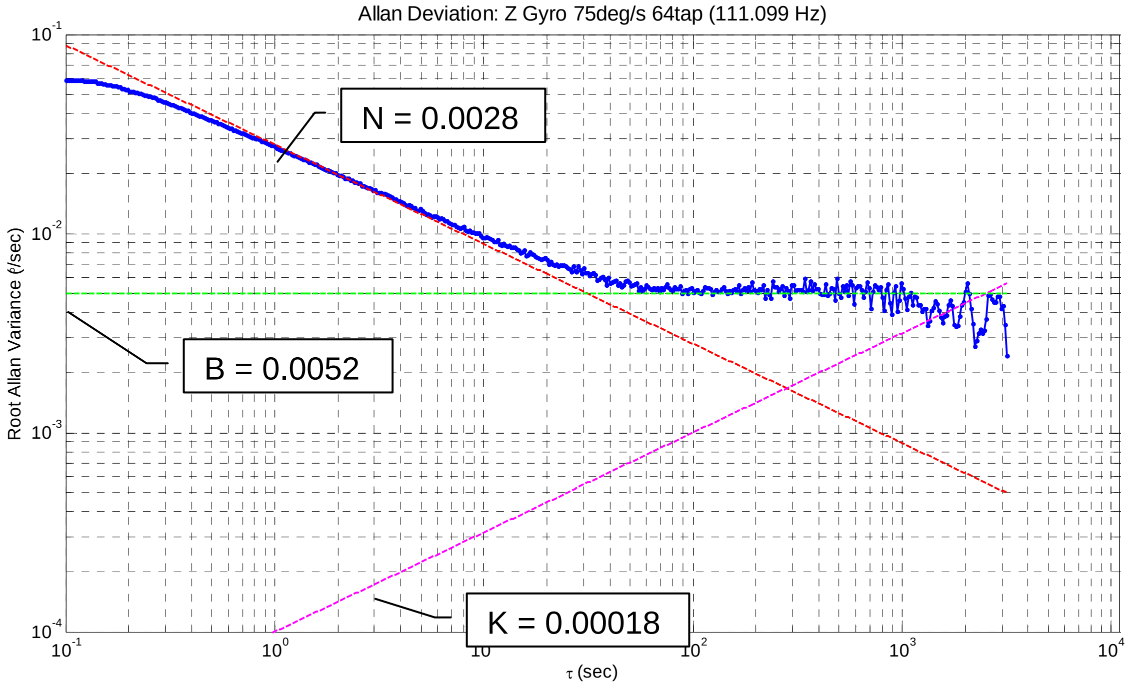


Figure 13. Calculated Root Allan Variance from Gyro Data 3.

The bias stability increased back to the value from Gyro Data 1 and the ARW stayed the same from Gyro Data 2. The shape of the graph, however, has changed. The RRW shown here is probably inaccurate. The +1/2 slope that is normally seen on the Allan variance is either very short or is in the uncertainty of the Allan variance calculation. The low-pass filtering appears to have reduced the RRW and possibly reduce other noises as well. It also is possible the increase in bias stability noise is due to the elimination of RRW. This change in noise source levels is not well understood. The -1/2 slope line for the ARW does not appear to fit as well as it did on the other Allan variance plots. This could mean that the ARW has been reduced slightly, at least for higher frequencies.

These three root Allan variance plots clearly show the expected outcome. The low-pass filtering has significantly reduced the noise of the gyro measurements. Table 11 shows a summary of the noise coefficients found from these three data sets.

	ARW (°/√sec)	Bias Stability (°/sec)	RRW (°/√sec^3)
Gyro Data 1	0.028	0.0052	0.00038
Gyro Data 2	0.028	0.0038	0.00018
Gyro Data 3	0.028	0.0052	0.00018[4]

Table 11. Summary of Noise Coefficients.

2. Verification of Gyro Model

The noise coefficients from Gyro Data 2 was used in the Simulink® model because of its realistic parameters and high confidence in the results. The simulated gyro noise and the tested gyro noise can be seen in Figure 14. The mean was subtracted from each data set to better compare the random noise levels. The Simulated and real noise are very similar, validating the Simulink® model.

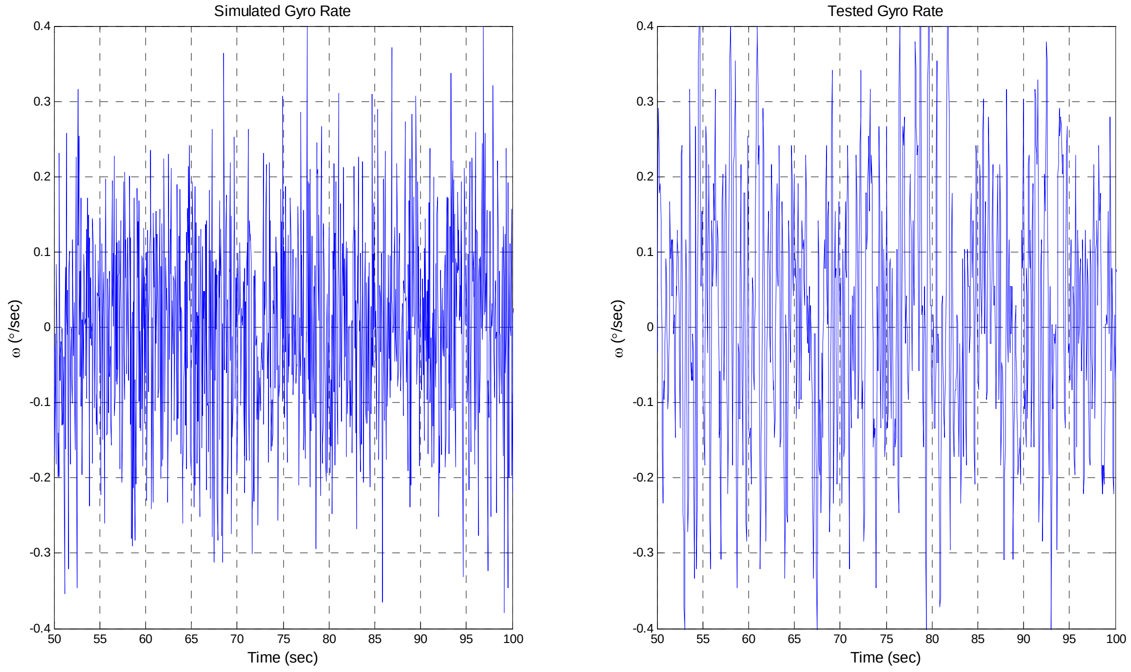


Figure 14. Simulated Gyro without Bias and Actual Gyro Noise.

---

[4] This number is questionable due to the poor quality of data.

3. Statistical Analysis of Magnetometer

The formulas and physical relationships have not been developed to use the Allan variance for a magnetometer. More mundane methods will be used to analyze the magnetometer noise. These will be the mean, to estimate the bias, the standard deviation and variance, to estimate the other noises. Two data sets at different sample rates were taken from the magnetometer. The statistics were calculated and the results are compared.

Figure 15 shows the statistics from Magnetometer Data 1. The mean (in green) is -161.1 mGauss, the 3σ (in red) or three times the standard deviation is 4.57 mGauss, and the variance is 2.32 mGauss. The graph on the left is a plot of the Z axis magnetometer over time and the graph on the right is a histogram of the same data to confirm the calculated statistics.

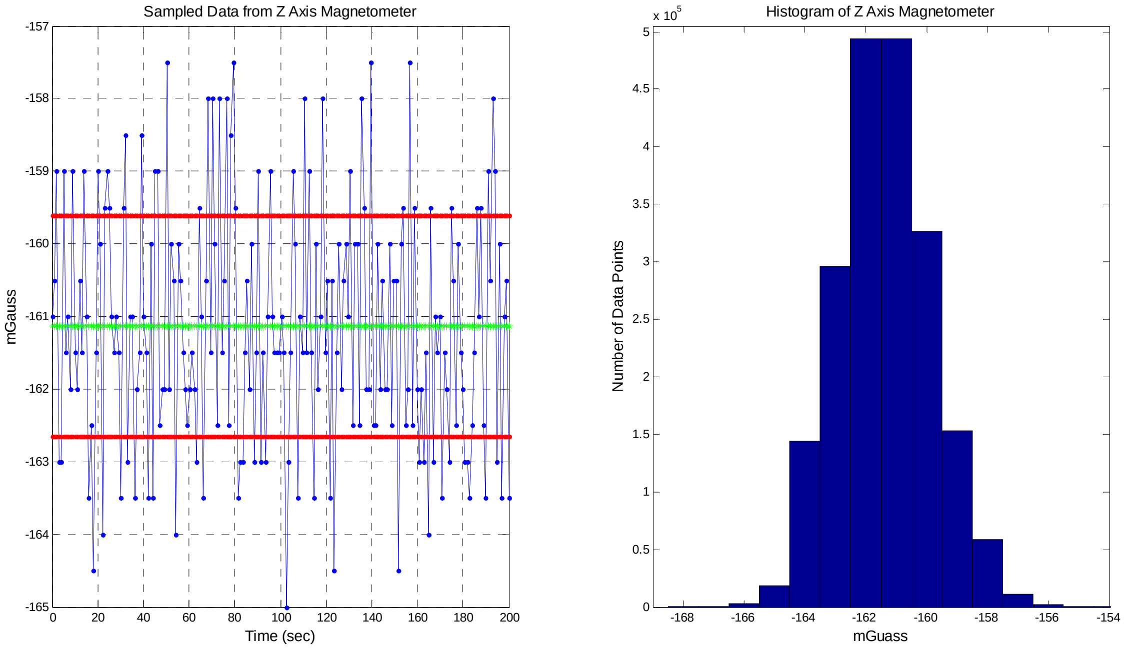


Figure 15. Statistics Graphs from Magnetometer Data 1.

Figure 16 shows the statistics from Magnetometer Data 2. The mean (in green) is -148.9 mGauss, the 3σ (in red) or three times the standard deviation is 4.39 mGauss, and the variance is 2.146 mGauss. The graph on the left is a plot of the Z axis magnetometer over time and the graph on the right is a histogram of the same data to confirm the calculated statistics.

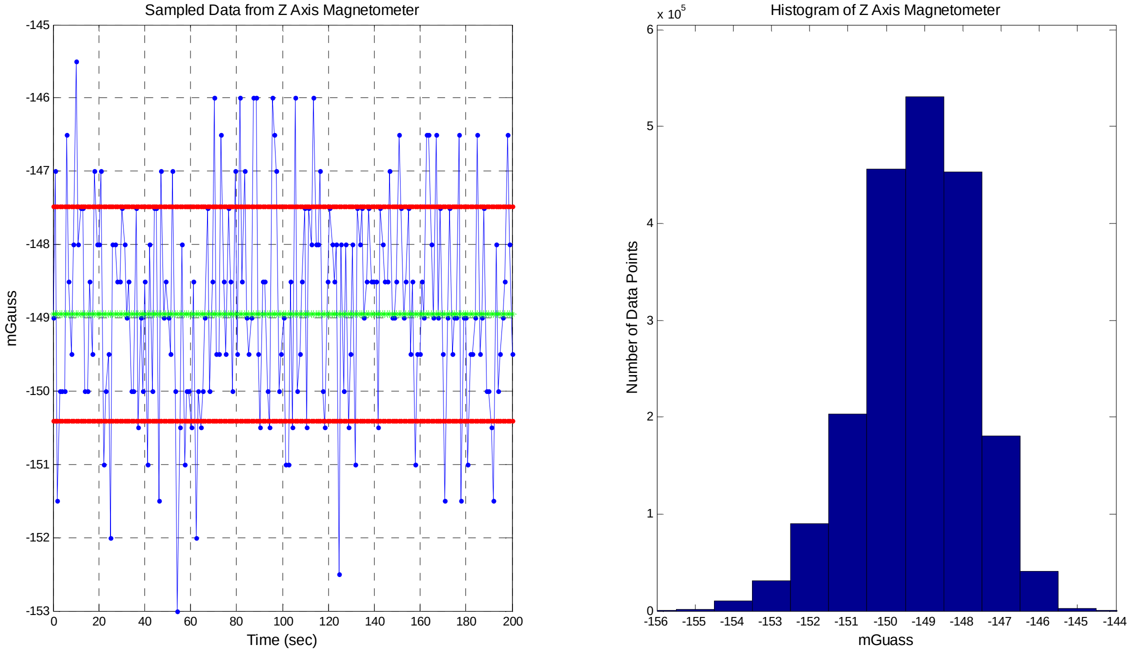


Figure 16. Statistics Graphs from Magnetometer Data 2.

The standard deviation and variance were very close for both data sets. The means, however, were significantly different. This is believed to be due to the test setup of the first data set. The IMU was placed too close to some metal washers and the ferrous material corrupted the absolute measurement of the magnetic field. This does not invalidate the data because it was constant throughout the test and therefore only changed the mean.

4. Verification of Magnetometer Model

Figure 17 shows the Simulink® model noise compared to the recorded noise. The simulated noise has the same statistics as the two hardware data sets. This validates the Simulink® Magnetometer model.

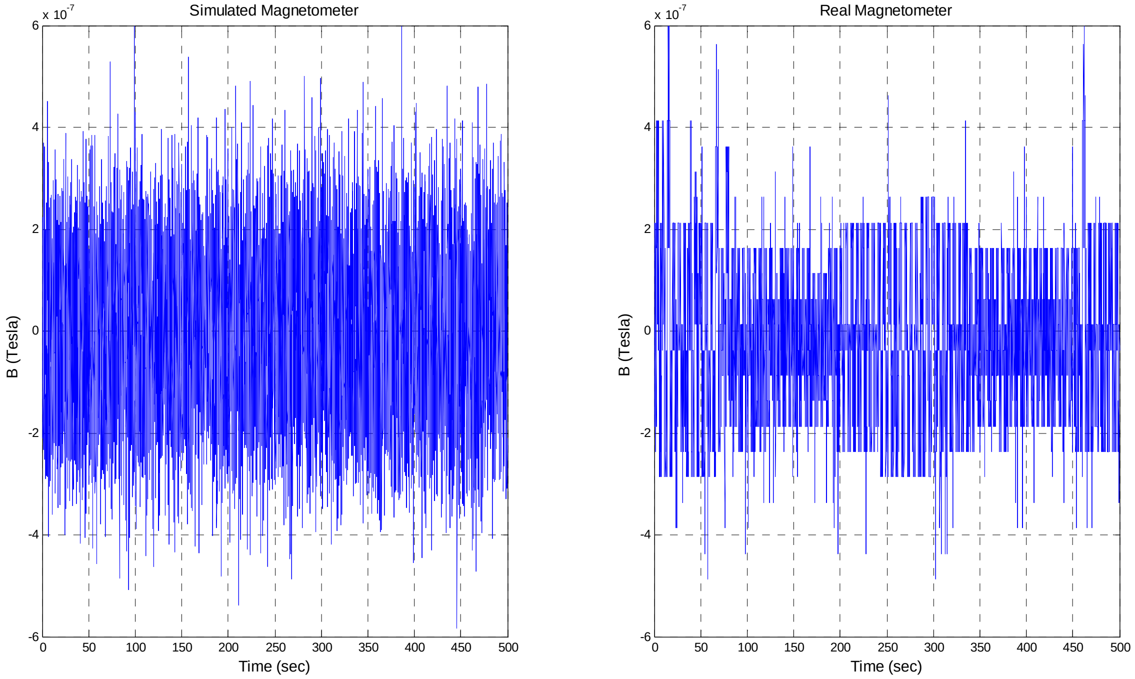


Figure 17. Simulated Magnetometer Noise and Actual Magnetometer Noise.

IV. NUMERICAL MODELING AND SIMULATION

A. PURPOSE

This chapter describes the general spacecraft model that was originally developed in [11] and has been subsequently further developed for this thesis. The notable additions include Earth's magnetic field model, attitude sensors models, and the Extended Kalman Filter described in Chapter VI. The model is run from a MATLAB® script file to set initial conditions and plot the results. The Simulink® model is described first and focuses on the additions to the model. The TINYSCOPE Simulink® Model and MATLAB® script files can all be seen in the Appendix.

B. SIMULINK MODEL

1. Orbital Propagation

The standard Euler's Equation(2.1) were used to create the orbital propagation block. The position vector **R**, the velocity vector **V**, were outputs for use in other blocks in the model. This double integrator system allows for the propagation of any orbit by altering the position and velocity initial conditions. This flexibility is very useful when analyzing a wide variety of possible orbits.

$$\ddot{\mathbf{R}} = -\frac{\mu}{r^3}\mathbf{R} \tag{2.1}$$

Other orbital elements are also computed here. The Beta angle (β), which is the angle between the sun line and the subsolar point (SSP), and the True Anomaly (ν), which is the angle between the position vector and periapsis, are two of these. They are calculated with Equations (2.2) and (2.3). The non-orbital elements, latitude and longitude are computed here as well for convenience. The eccentricity (*e*) and inclination (*i*) of the orbit are predefined and constant for the purposes of the simulation.

$$\beta = \sin^{-1}\left(\sin u \sin\gamma \cos i + \cos u \sin i \sin\omega - \sin u \cos\gamma \sin i \cos\omega\right) \qquad (2.2)$$

Where the right ascension of the sun in the ecliptic plane is $u = u_0 + \dot{u}t$, $\dot{u} = 0.985648°$/mean solar day, the Earth spin axis tilt with respect to the ecliptic plane is $\gamma = 23.442°$, and the right ascension of the line of nodes is $\omega = \omega_0 + \dot{\omega}t$, $\dot{\omega} = \left(-9.9639\rho^{3.5}\cos i\right)/\left(1-e^2\right)^2$ °/mean solar day. Where the Earth angular radius is $\rho = \sin^{-1}\left(R_e/\left(R_e+h\right)\right)$.

$$\nu = \cos^{-1}\left(\frac{\mathbf{E}\cdot\mathbf{R}}{er}\right) \qquad (2.3)$$

where, $\mathbf{E} = \left(\left(v^2 - \mu/r\right)\mathbf{R} - \left(\mathbf{R}\bullet\mathbf{V}\right)\mathbf{V}\right)/\mu$ and r is the magnitude of the position vector of the satellite.

2. Environmental Effects

This block models the space environment that the satellite will be experiencing. It uses models developed from commonly known relationships. There are three aspects of the space environment were modeled; they are: the Earth's Magnetic field, the Sun, and the Earth's atmospheric density.

a. Earth's Magnetic Field Model

The Earth's magnetic field can be approximated by a magnetic dipole. This model is not very accurate because the true magnetic field has many variations due to interactions from within the Earth as well as from outer space. The dipole is, however, much easier to calculate and is accurate enough for the purposes of these preliminary simulations. Equation(2.4)[5] is the vector form of the magnetic dipole where *m* is the Earth's vector dipole moment, μ_0 is the permeability of free space, and $\vec{\mathbf{r}}$ is the satellite position unit vector. Because the Earth's magnetic field is displaced by approximately 11.7° from the Earth's

---

[5] The delta function is zero for the magnetic field except at the origin of the dipole. Since the spacecraft would never be at the center of the Earth, this term was neglected.

axis, the dipole moment was rotated into the ECEF frame. The transformation matrix used was Equation(2.5). The position vector also had to be rotated from ECI to ECEF in Equation(2.6) using the Earth's rotation rate, $\varphi = -7.2921\text{e-}005$ rad/sec.

$$\mathbf{B} = \frac{\mu_0}{4\pi r^3}\left(3\left(m \bullet \bar{\mathbf{r}}\right)\bar{\mathbf{r}} - m\right) + \frac{2\mu_0}{3} m\delta^3\left(r\right) \tag{2.4}$$

$$^{ecef}_{mag}T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{2.5}$$

$$^{ecef}_{eci}T = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.6}$$

Another option tried was the World Magnetic Model 2005 from the Aerospace Blockset of MATLAB/Simulink® for its high fidelity. It takes the altitude, the latitude, and the longitude of the spacecraft as inputs, all of which were derived from the position vector produced by the orbit propagator block. The output is the magnetic field vector in Geodetic coordinates. Unfortunately, this model is too computationally intensive to be practical in the current implementation of the spacecraft model.

b. Atmospheric Density

A look-up table using atmospheric density data points from 100 to 700 km approximates the nominal atmospheric density, ρ_{ao}. This density is used as a starting point for Equation(2.7). This equation takes into account the differences in density due to the sun and eclipse. It also accounts for the beta angle. The initial density is chosen based on the nominal altitude and the expected solar activity.

$$\rho_a = \rho_{ao} 1.5^{(\cos\beta\cos\nu)} \tag{2.7}$$

c. Solar Simulation

The sun is simulated using two well known equations that describe the Sun Direction Vector of the body with respect to the orbital frame, Equation(2.8), and when the satellite is in eclipse, Equation(2.9).

$$\hat{\mathbf{S}}_\mathbf{b} = C_{BO} \begin{bmatrix} \cos\beta \sin\nu \\ \sin\beta \\ \cos\beta \cos\nu \end{bmatrix} \tag{2.8}$$

$$\cos\nu + \frac{\cos\rho}{\cos\beta} < 0 \tag{2.9}$$

3. Dynamics and Kinematics

This block uses standard equations to calculate the rigid body dynamics as well as the quaternion kinematics.

a. Dynamics

The Euler Equation(2.10) is used to calculate the nonlinear dynamics of the spacecraft. These equations relate the applied torques to the spacecraft angular rates. Only spacecraft rigid body dynamics are accounted for and all forces are measured along the three principle axes.

$$T = \begin{cases} J_x\dot{\omega}_x + \left(J_z - J_y\right)\omega_z\omega_y \\ J_y\dot{\omega}_y + \left(J_x - J_z\right)\omega_x\omega_z \\ J_z\dot{\omega}_z + \left(J_y - J_x\right)\omega_y\omega_z \end{cases} \tag{2.10}$$

b. Kinematics

This is the standard quaternion Equation(2.11) that relates the angular velocity and the current quaternion to produce the derivative of the quaternion. This can then be integrated to produce the propagated quaternion.

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{2.11}$$

4. Disturbance Torques

These blocks calculate the three primary disturbance torques. They make heavy use of the environmental blocks outputs discussed earlier.

a. Gravity Gradient Torque

This disturbance torque is due to gravity acting on the spacecraft. Variations in mass occur throughout any satellite and therefore gravity acts more strongly on certain parts of the spacecraft than others. This gradient in strength of gravitational force causes a torque on the spacecraft body. The standard equations for the gravity gradient disturbance torque are:

$$T_{GG} = \frac{3\mu}{R^3}\begin{bmatrix} \left(J_z - J_y\right)c_2c_3 \\ \left(J_x - J_z\right)c_1c_3 \\ \left(J_y - J_x\right)c_1c_2 \end{bmatrix} \quad \text{where,} \quad \begin{Bmatrix} c_1 \\ c_2 \\ c_3 \end{Bmatrix} = C_{BO}\begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} \qquad (2.12)$$

b. Aerodynamic Torque

This disturbance torque is due to the inelastic impacts of molecules in the upper atmosphere onto the spacecraft. These impacts transfer momentum from the molecule to the spacecraft causing a drag force. The net drag force acts on the center of pressure located on the spacecraft face in the direction of travel. The equation describing this torque on a spacecraft assuming flat surfaces is:

$$T_{aero} = -\rho_a V_R^2 \hat{\mathbf{V}}_R \times \sum_{i=1}^{N}\left(\vec{\mathbf{c}}_{pi}\left(\hat{\mathbf{V}}_R \cdot \hat{\mathbf{n}}_i\right)A_i\right)H\left(\hat{\mathbf{V}}_R \cdot \hat{\mathbf{n}}_i\right) \qquad (2.13)$$

where, $\vec{\mathbf{c}}_{pi} = \vec{\mathbf{R}}_{CPi} - \vec{\mathbf{R}}_{CM}$ and $\hat{\mathbf{V}}_R$ is a unit vector along relative velocity of atmosphere WRT the spacecraft.

$$H\left(\hat{\mathbf{V}}_R \cdot \hat{\mathbf{n}}_i\right) = \begin{cases} 1, & \hat{\mathbf{V}}_R \cdot \hat{\mathbf{n}}_i > 0 \\ 0, & \hat{\mathbf{V}}_R \cdot \hat{\mathbf{n}}_i \le 0 \end{cases}$$

c. Solar Torque

This disturbance torque is very similar to the aerodynamic torque. The major difference is that the force acting on the spacecraft is due to the impingement of light from the sun. The sun vector calculated in Equation(2.14) is used for this calculation.

$$T_{solar} = -\rho_s \hat{\mathbf{S}}_b \times \sum_{i=1}^{N} \left(\hat{\mathbf{c}}_{pi} \left(\hat{\mathbf{S}}_b \cdot \hat{\mathbf{n}}_i \right) A_i \right) H_s \left(\hat{\mathbf{S}}_b \cdot \hat{\mathbf{n}}_i \right) \tag{2.14}$$

where,

$$\vec{\mathbf{c}}_{pi} = \vec{\mathbf{R}}_{CPi} - \vec{\mathbf{R}}_{CM}$$

$$H\left(\hat{\mathbf{S}}_b \cdot \hat{\mathbf{n}}_i \right) = \begin{cases} 1, & \hat{\mathbf{S}}_b \cdot \hat{\mathbf{n}}_i > 0 \\ 0, & \hat{\mathbf{S}}_b \cdot \hat{\mathbf{n}}_i \le 0 \end{cases}$$

5. Sensor Models for Simulations

This block contains four different types of sensors. A magnetometer, a gyroscope, a sun sensor, and a star tracker are simulated with noise sources. The goal was to create sensor simulations that would accurately mimic actual hardware.

a. Gyroscope

The gyroscope takes the angular velocity from the Euler Equations and adds noise to create a realistic measured $\tilde{\omega}$. Two types of noise are added to the signal, Angular Random Walk (ARW) and Bias. The ARW is a zero-mean Gaussian random noise with a variance determined from Chapter III. The bias is modeled as the integration of white noise with a variance called Rate Random Walk, also determined in Chapter III. The integrator is initialized at the initial bias of the hardware. These terms can be seen in mathematical model of the gyro from [24], Equation(2.15).

$$\begin{gathered} \tilde{\omega} = \omega + \beta + \xi_{sf} + \xi_{ma} + \eta_v \\ \dot{\beta} = \eta_u \end{gathered} \tag{2.15}$$

Where ω is the true angular rate, β is the bias, ξ_{sf} is gyro scale factor errors, ξ_{ma} is gyro misalignment errors, η_v is the ARW, and η_u is the RRW. This equation was the basis for the Simulink® model of the gyro seen in Figure 18.

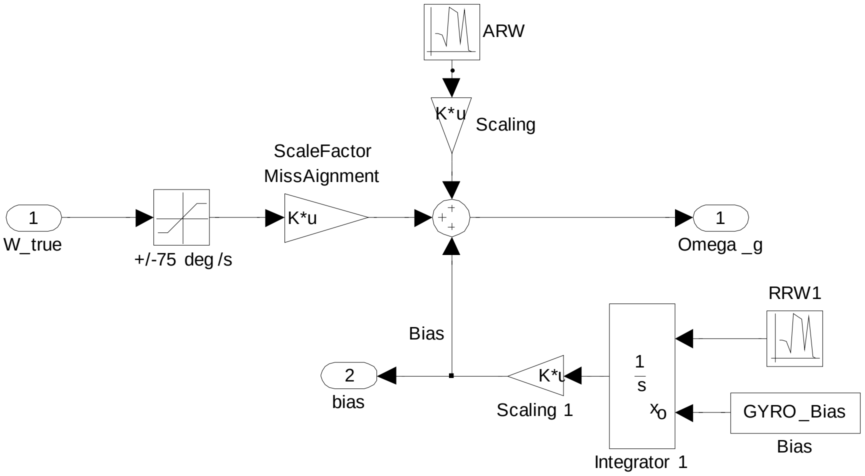


Figure 18. Realistic Simulink® Model for a Gyroscope.

The ARW, RRW, and initial gyro bias can easily be seen in the model. The gyro noise sources are "scaled" by $T_s^{-0.5}$ like in [24]. This method uses the sample rate, T_s , to correct the units of ARW (°/√s) and RRW (°/√s^3) to °/s and °/s^2 respectively. The result produces realistic noise in the model. The misalignment inaccuracies are added with a gain block in the model. The gain is $I_{3x3} + G$, where the diagonal values of G are the percent error in scale factor and the off-diagonal values of G are the percent error of misalignment. The gyroscope model also simulates the dynamic range of the hardware with a saturation block. These all combined to create a realistic measurement ($\tilde{\omega}$) to be used in the simulation.

b. Magnetometer

The magnetometer takes the B-field in the body frame, B_b, adds realistic noise to simulate the actual magnetometer hardware characteristics. Equation(2.16) describes the model used as the basis for the Simulink® model in

Figure 19. One noise source is added to B_b, the output noise level, ε, of the magnetometer. The specification sheet lists the output noise in rms, this is equivalent to the 1σ value of the noise. The scale factor, ξ_{sa}, and misalignment, ξ_{ma}, error were also added into the model.

$$B_m = B_b + \xi_{sf} + \xi_{ma} + \varepsilon \tag{2.16}$$

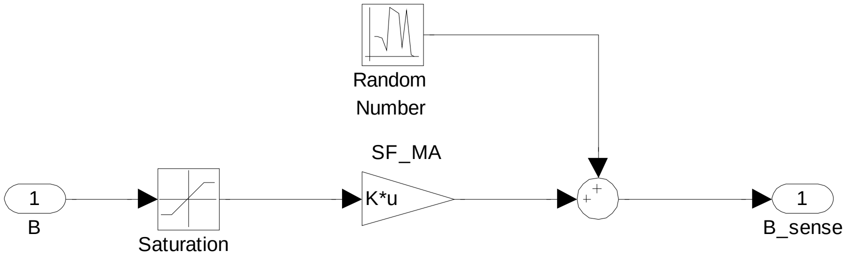


Figure 19. Realistic Simulink® Model for a Magnetometer.

The magnetometer simulates other physical characteristics of the hardware. A saturation block is used to limit the range. The misalignment and scale factor inaccuracies are added with a gain block using the same equation as the gyro model. These all combined create a realistic measured B_b to be used in the simulation.

c. Sun Sensor

The sun sensor does not add a Gaussian random noise to the measured signal because, as stated in Chapter III, the dominate noise is special in nature. Also note there is no bias for a line-of-sight sensor. The special noise was approximated with a modified Bessel function, Equation(2.17). The important things to model were the wave like structure of the noise and the rms error of 0.1 °. The result of the Bessel function was stored in a look-up table indexed by the coordinate of the Sun vector on the simulated sensor detector. The small angle approximation allows the 0.1 °rms error to be directly added to the measured Sun vector. The other part of this simulation is determining if the

Sun is in view of the sensor. First, it is determined if the spacecraft is in eclipse. This was described earlier in the Environmental Effects section. Next, the sun vector must be changed into the body coordinates, S_b. This is done by multiplying by C_{BN}. Then it is rotated into the individual sensors frame. Two sun sensors where simulated in this model, but more could easily be included. The two sun sensors frames are set to maximize the time the sun will be seen by at least one sensor. Then it can be determined if the Sun is in the sensor's field-of-view. This is done by ensuring the S_b is on the correct side if the spacecraft and is in the sensor's field-of-view as in Equation(2.18).

$$I_v(r) = \left(\frac{r}{2}\right)^v \sum_{k=0}^{\infty} \left\{ \frac{\left(\frac{r^2}{4}\right)^k}{k!\Gamma(v+k+1)} \right\} \cos(\theta) \qquad (2.17)$$

Where Γ is the gamma function, *v* is a real constant, and *r* and *θ* are the radius and angle in polar coordinates respectively.

$$\bar{S}_x < -\sqrt{1-FOV^2}, \quad FOV = \sqrt{\delta_x^2 + \delta_y^2} \qquad (2.18)$$

Where δ is the location on the detector and the – denotes a unit vector.

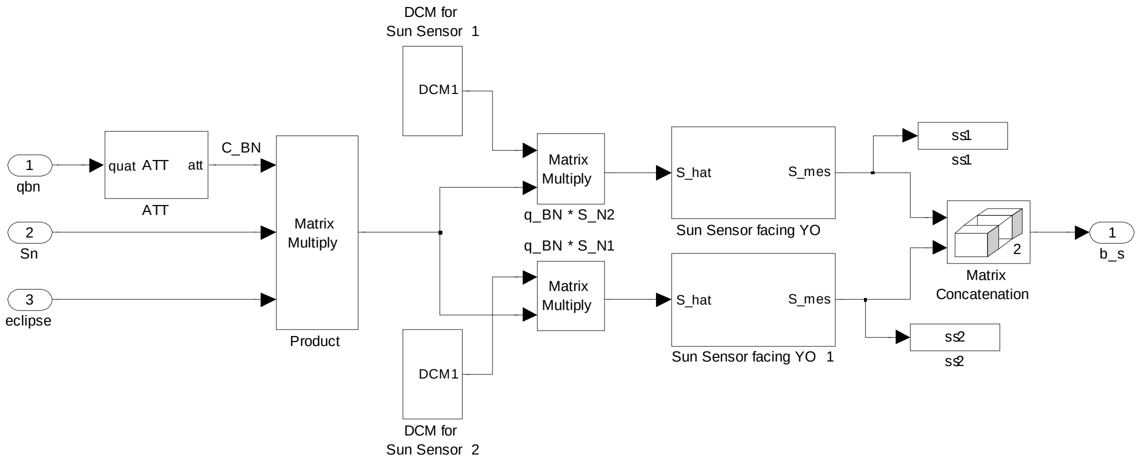


Figure 20. Realistic Simulink® Model for Two Sun Sensors.

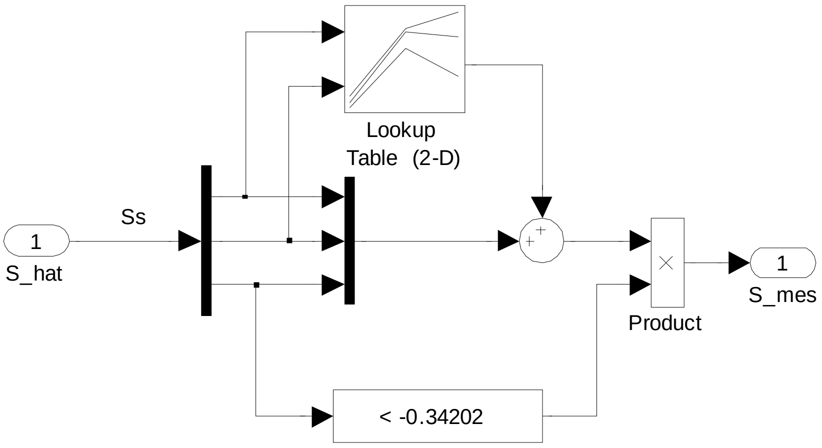


Figure 21. One Sun Sensor Facing Block.

d. Star Tracker

The star tracker model used here is a much simpler one than in other simulations. This is because the EKF developed here will assume the star tracker is a ”black box.” That is to say that the EKF will only be a user of the star tracker solution and not be an integral part of the star tracker itself[6]. This will allow the EKF to work with any COTS star tracker that might be developed later.

The star tracker will output a quaternion as many star trackers do, however, it will use Euler angles as an input. This allows for easy application of the known error of the star tracker to the measured data. A Gaussian random noise with a variance of ±70 arcsec is added to each Euler angle. The angles are then used to calculate the measured quaternion.

---

[6] The EKF can be easily modified to be an integral part of a star tracker. The EKF developed in [27], which this EKF relies heavily on, does just this.

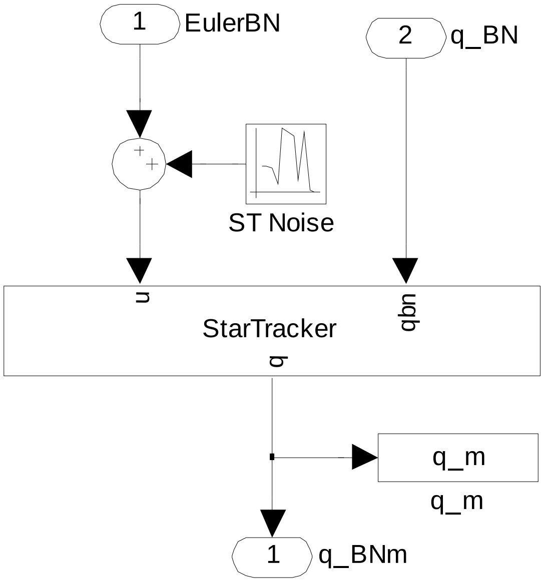


Figure 22. Realistic Simulink® Model for Star Tracker.

6. Gain Scheduled Quaternion Feedback Controller

This block was developed in [11] and has not been modified for this thesis. No simulations will be done using the controller so it will not be included in this description.

C. MATLAB® CODE

The Simulink® Model used a number of MATLAB® M-files to both initialize the simulation and perform calculations. Files that were not developed for this thesis will not be explained here. All the code used for this thesis is in the Appendix.

1. TINYSCOPE Main Script

The TINYSCOPE Simulink® Model is initialized and run from *TinyscopeMainScript.* This file has all the constants and parameters to run the simulation in it. It also calls functions to perform specific calculations.

2. Euler to Quaternion

This script is a function that is called both in *TinyscopeMainScript* and in various blocks of the TINYSCOPE Simulink® Model. It takes the Euler angle vector and calculates the 3-2-1 sequence quaternion.

3. Quat2Euler

This script is used to convert a quaternion to an Euler angle in the 3-2-1 sequence.

4. Calculate 6U Spacecraft

This script was developed in [11]. It generates the moment of inertia, center of gravity, and center of pressure for the TINYSCOPE model.

5. Plotting Functions

Several plotting function were developed to aid in presenting the simulation results. These functions take in simulation data and generate a series of plots that display the data in a usable fashion. Plotting functions developed here include:

- PlotOrbit
- PlotMeasurements
- PlotEKFErrors

V. KALMAN FILTERING APPROACH TO STATE ESTIMATION

A. BACKGROUND

The Kalman Filter is a method to recursively estimate the state vector using stochastic processes. The filter finds the optimum solution by minimizing the mean square error of the estimated state vector with a system model of the plant dynamics and sensor noise. R. E. Kalman first developed this method in 1960 [25]. Since then it has been used in multiple disciplines ranging from signal processing to spacecraft control. It has also been expanded upon and further developed many times to now include very good nonlinear state estimators.

The Kalman Filter uses a two-step process of predicting the state vector using a system model and updating the state vector using measurements (see Figure 23). This means that only the previous state estimate need be stored until the next time step. Thus, the recursive nature of the Kalman filter makes it easily implemented on a digital computer.

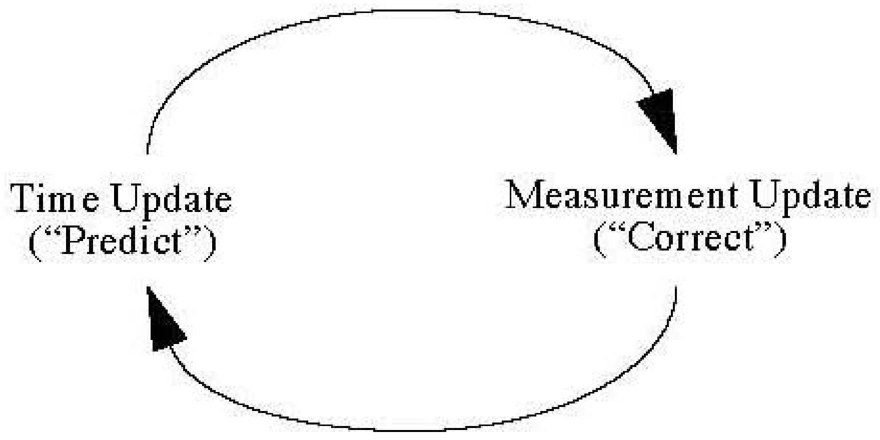


Figure 23. Kalman Filter "Predict-Correct" Cycle. From [26].

This also leads to the advantage that multiple sensors of different types can be used to update the estimated state vector. Through an error covariance matrix, the confidence level of each measurement is tracked to appropriately weight the update. When a low accuracy measurement is used to update the state vector, it can be weighted lower than the predicted model; while a very accurate measurement is weighted much more heavily then the predicted model. The result is an estimation that when properly implemented can provide more accurate state estimation then the direct measurements alone.

B. DISCRETE EXTENDED KALMAN FILTER

This section will show the structure of the Kalman Filter. Specifically a formulation using discrete nonlinear plant dynamic models with discrete sensors will be shown. This formulation is of interest because the implementation of the EKF developed will be on a microprocessor. The digital nature of the microprocessor lends itself to discrete functions. So, the spacecraft dynamics and kinematics are approximated with discrete nonlinear equations and the real sensors are discrete digital sensors. The derivations have been done in numerous publications but for the purposes of consistency and simplicity all of these equations are from [27].

To begin, the continuous nonlinear model and measurements are defined as:

$$\dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t), t\big) + G(t)\mathbf{w}(t) \tag{3.1}$$

$$\mathbf{y}(t) = \mathbf{h}\big(\mathbf{x}(t), t\big) + \mathbf{v}(t) \tag{3.2}$$

Then using the first-order approximation of the nonlinear system dynamics **f** and nonlinear measurement **h** continuous Riccati equations with Equation(3.3), we can approximate the discrete Riccati equation with a Taylor-series expansion for exp(FT_s) to the second term in Equation(3.4):

$$F(\hat{\mathbf{x}}, t) \equiv \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}} \qquad H(\hat{\mathbf{x}}, t) \equiv \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}} \tag{3.3}$$

$$\Phi_k \approx I + FT_s \tag{3.4}$$

Applying Equations (3.3) and (3.4) to Equations (3.1) and (3.2) leads to the discrete nonlinear system dynamic and measurement equations:

$$\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + \Gamma_k \mathbf{u}_k + \Upsilon_k \mathbf{w}_k \tag{3.5}$$

$$\mathbf{y}_k = \mathbf{h}_k\left(\mathbf{x}_k\right) + \mathbf{v}_k \tag{3.6}$$

Where $\mathbf{w}_\mathrm{k}$ & $\mathbf{v}_\mathrm{k}$ are zero-mean Gaussian noise process with covariances given by the expectation equations:

$$E\left\{\mathbf{w}_k \mathbf{w}_j^T\right\} = \begin{cases} 0, k \neq j \\ Q_k, k = j \end{cases} \tag{3.7}$$

$$E\left\{\mathbf{v}_k \mathbf{v}_j^T\right\} = \begin{cases} 0, k \neq j \\ R_k, k = j \end{cases} \tag{3.8}$$

$$E\left\{\mathbf{v}_k \mathbf{w}_k^T\right\} = 0 \tag{3.9}$$

The Q_k matrix accounts for state process noise while the R_k matrix accounts for expected measurement noise.

The current state will be propagated by estimating the truth model of Equation(3.5) with:

$$\hat{\mathbf{x}}_{k+1}^- = \Phi_k \hat{\mathbf{x}}_k^+ + \Gamma_k \mathbf{u}_k \tag{3.10}$$

The current state will be updated with a measurement, $\mathbf{y}_\mathbf{k}$, in:

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k\left[\mathbf{y}_k - H_k \hat{\mathbf{x}}_k^-\right] \tag{3.11}$$

The gain K_k changes with time properly weighting the relative confidence of the accuracy of the propagated state verses the measured state. To find K_k first the state error and error covariance matrixes must be defined:

$$\begin{aligned} \tilde{\mathbf{x}}_k^- &= \hat{\mathbf{x}}_k^- - \mathbf{x}_k \quad & \tilde{\mathbf{x}}_{k+1}^- &= \hat{\mathbf{x}}_{k+1}^- - \mathbf{x}_{k+1} \\ \tilde{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^+ - \mathbf{x}_k \quad & \tilde{\mathbf{x}}_{k+1}^+ &= \hat{\mathbf{x}}_{k+1}^+ - \mathbf{x}_{k+1} \end{aligned} \tag{3.12}$$

$$\begin{aligned} P_k^- &\equiv E\left\{\tilde{\mathbf{x}}_k^- \tilde{\mathbf{x}}_k^{-T}\right\} \qquad & P_{k+1}^- &\equiv E\left\{\tilde{\mathbf{x}}_{k+1}^- \tilde{\mathbf{x}}_{k+1}^{-\ T}\right\} \\ P_k^+ &\equiv E\left\{\tilde{\mathbf{x}}_k^+ \tilde{\mathbf{x}}_k^{+T}\right\} \qquad & P_{k+1}^+ &\equiv E\left\{\tilde{\mathbf{x}}_{k+1}^+ \tilde{\mathbf{x}}_{k+1}^{+\ T}\right\} \end{aligned} \tag{3.13}$$

Substituting Equations (3.5) and (3.10) into Equation(3.12) and substituting the resulting equation into Equation(3.13) leads to:

$$P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + \Upsilon_k Q_k \Upsilon_k^T \tag{3.14}$$

Because $\mathbf{w}_k$ and $\tilde{\mathbf{x}}_k^+$ are uncorrelated the terms $E\left\{\mathbf{w}_k\tilde{\mathbf{x}}_k^{+T}\right\} = E\left\{\tilde{\mathbf{x}}_k^+\mathbf{w}_k^T\right\} = 0$. To find the updated error covariance matrix, substitute Equation(3.6) into Equation(3.11). Then substitute the resulting equation into Equation(3.12) and reduce, leads to:

$$P_k^+ = \left[I - K_k H_k\left(\hat{\mathbf{x}}_k^-\right)\right]P_k^- \tag{3.15}$$

In order to actually calculate the gain *K*, the trace of the updated error covariance matrix must be minimized. Solving gives:

$$K_k = P_k^- H_k^T\left(\hat{\mathbf{x}}_k^-\right)\left[H_k\left(\hat{\mathbf{x}}_k^-\right)P_k^- H_k^T\left(\hat{\mathbf{x}}_k^-\right) + R_k\right]^{-1} \tag{3.16}$$

The Continuous-Discrete Extended Kalman Filter is summarized in Table 12. The initialization of the EKF is important because the estimated state is assumed to be close to the true state. Large initial condition errors on some nonlinear plants can cause instabilities in the EKF. This can be avoided through simulation and testing.

	Discrete Extended Kalman Filter
Model	$\dot{\mathbf{x}}_{k+1} = \Phi_k\mathbf{x}_k + \Gamma_k\mathbf{u}_k + \Upsilon_k\mathbf{w}_k$ $\mathbf{y}_k = H_k\left(\mathbf{x}_k\right) + \mathbf{v}_k$
Initialize	$\hat{\mathbf{x}}\left(t_0\right) = \hat{\mathbf{x}}_0$ $P_0 = E\left\{\tilde{\mathbf{x}}\left(t_0\right)\tilde{\mathbf{x}}^T\left(t_0\right)\right\}$
Gain	$K_k = P_k^- H_k^T\left(\hat{\mathbf{x}}_k^-\right)\left[H_k\left(\hat{\mathbf{x}}_k^-\right)P_k^- H_k^T\left(\hat{\mathbf{x}}_k^-\right) + R_k\right]^{-1}$
Update	$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K\left[\mathbf{y}_k - \mathbf{h}\left(\hat{\mathbf{x}}_k^-\right)\right]$ $P_k^+ = \left[I - K_k H_k\left(\hat{\mathbf{x}}_k^-\right)\right]P_k^-$
Propagation	$\hat{\mathbf{x}}_{k+1}^- = \Phi_k\hat{\mathbf{x}}_k^+ + \Gamma_k\mathbf{u}_k$ $P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + \Upsilon_k Q_k \Upsilon_k^T$

Table 12. Discrete Extended Kalman Filter. From [27].

C. CHALLENGES OF MULTIPLE SENSOR SYSTEM

In the formulation of the above EKF, the sensors are modeled as discrete measurements. This is a very good model for digital sensors or sampled analog sensors. This type of sensor is ideal to be used with a Kalman filter implemented on the microprocessor. There is, however, an inherent problem. Different types of sensors produce solutions at different rates. This causes a problem for a Kalman filter that has to have all of the measurements available to update the state vector. A solution to this is to use superposition for the updates. This is possible for an EKF because although it is modeling a nonlinear system it linearizes the propagation and update equations about the current state estimate. This technique of using superposition, was first suggested by James Murrell in [28]. It has since been applied many times and in different ways.

The essence of the technique is to update the gain, error covariance, and state error vector with each successively available measurement. Once all of the measurements have been taken into account, the EKF will propagate the estimated state and covariance matrix until the next measurement or set of measurements are available. Interestingly, this also greatly reduces the computational burden. Instead of calculating a gain matrix that requires an inverse of a 3n x 3n matrix, only a 3 x 3 matrix inverse is required n times with Murrell’s version.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. IMPLEMENTATION OF EXTENDED KALMAN FILTER FOR MULTI-RATE SENSORS

The Discrete Extended Kalman Filter discussed in Chapter IV will form the bases of the EKF developed for this thesis. The EKF developed here will use ideas and equations from several sources. These include [14], [27],[29], and [30]. The combination of these different sources helps create an EKF that can handle data from gyros, star trackers, sun sensors, and/or magnetometers in an efficient way. The goal of this EKF is to produce an accurate attitude estimates using a gyro and any combination of other sensors. It will also be computationally efficient for implementation on a low power microprocessor.

A. MUTIPLICATIVE QUATERNION EXTENDED KALMAN FILTER

Because this EKF will use different types of sensors running at different rates and it will be implemented with on-board computation, a version of Murrell's form briefly described in Chapter IV was used. The basic structure and much of the derivation of this EKF comes from [27]. The significant changes occur in the formulation of the observation matrix H_k and the calculation of the residual, ε , for each of the sensor types. To begin the derivation the state vector, Equation(4.1), was chosen to be composed of four quaternion elements and three gyro bias elements.

$$\mathbf{X} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & \beta_x & \beta_y & \beta_z \end{bmatrix}^T \tag{4.1}$$

Where the quaternion is defined by $\mathbf{q} = \begin{bmatrix} \hat{\mathbf{e}} \sin(\alpha / 2) & \cos(\alpha / 2) \end{bmatrix}^T = \begin{bmatrix} \varsigma & q_4 \end{bmatrix}^T$ and follows the normalization $\mathbf{q}^T \mathbf{q} = 1$. The normalization constraint prevents a simple calculation of the quaternion error by subtraction. A different approach called the multiplicative error quaternion must be calculated. This is defined as

$$\delta\mathbf{q} = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1} \tag{4.2}$$

Where $\delta\mathbf{q} = \begin{bmatrix} \delta\varsigma & \delta q_4 \end{bmatrix}^T$ and the inverse quaternion is $\mathbf{q}^{-1} = \begin{bmatrix} -\varsigma & q_4 \end{bmatrix}^T$. Taking the time derivative yields:

$$\delta\dot{\mathbf{q}} = \dot{\mathbf{q}} \otimes \hat{\mathbf{q}}^{-1} + \mathbf{q} \otimes \dot{\hat{\mathbf{q}}}^{-1} \tag{4.3}$$

This eventually gives the estimated quaternion kinematics[7]

$$\dot{\hat{\mathbf{q}}} = \frac{1}{2}\Xi\left(\hat{\mathbf{q}}\right)\hat{\omega} = \frac{1}{2}\Omega\left(\hat{\omega}\right)\hat{\mathbf{q}} \tag{4.4}$$

$$\Xi\left(\mathbf{q}\right) \equiv \begin{bmatrix} q_4 I_{3x3} + \left[\varsigma \times\right] \\ -\varsigma^T \end{bmatrix} \qquad \Omega\left(\omega\right) \equiv \begin{bmatrix} -\left[\omega \times\right] & \omega \\ -\omega^T & 0 \end{bmatrix} \tag{4.5}$$

Now to find the discrete propagation of the quaternion kinematics Equation(4.4) will be expanded using the power series approach in Equation(4.6).

$$e^{\frac{\Omega(\hat{\omega})t}{2}} = \sum_{k=0}^{\infty} \left\{ \frac{\left[\frac{1}{2}\Omega\left(\hat{\omega}\right)t\right]^{2k}}{\left(2k\right)!} + \frac{\left[\frac{1}{2}\Omega\left(\hat{\omega}\right)t\right]^{2k+1}}{\left(2k+1\right)!} \right\} \tag{4.6}$$

Next substituting the identities $\Omega^{2k}\left(\hat{\omega}\right) = \left(-1\right)^k \left\|\hat{\omega}\right\|^{2k} I_{4x4}$ and $\Omega^{2k+1}\left(\hat{\omega}\right) = \left(-1\right)^k \left\|\hat{\omega}\right\|^{2k} \Omega\left(\hat{\omega}\right)$ into Equation(4.6) gives

$$e^{\frac{\Omega(\hat{\omega})t}{2}} = I_{4x4} \sum_{k=0}^{\infty} \frac{\left(-1\right)^k \left[\frac{1}{2}\left\|\hat{\omega}\right\| t\right]^{2k}}{\left(2k\right)!} + \left\|\hat{\omega}\right\|^{-1} \Omega\left(\hat{\omega}\right) \sum_{k=0}^{\infty} \frac{\left(-1\right)^k \left[\frac{1}{2}\left\|\hat{\omega}\right\| t\right]^{2k+1}}{\left(2k+1\right)!} \tag{4.7}$$

Then simplifying with the cosine and sine functions leads to Equation(4.8).

$$e^{\frac{\Omega(\hat{\omega})t}{2}} = I_{4x4} \cos\left(\frac{1}{2}\left\|\hat{\omega}\right\| t\right) + \Omega\left(\hat{\omega}\right) \frac{\sin\left(\frac{1}{2}\left\|\hat{\omega}\right\| t\right)}{\left\|\hat{\omega}\right\|} \tag{4.8}$$

Now the quaternion propagation can be simply written as

$$\hat{\mathbf{q}}_{k+1}^{-} = \bar{\Omega}\left(\hat{\omega}_k^{+}\right)\hat{\mathbf{q}}_k^{+} \tag{4.9}$$

Where $\hat{\omega}_k^{+}$ and $\hat{\mathbf{q}}_k^{+}$ are the post-update estimates and $\bar{\Omega}\left(\hat{\omega}_k^{+}\right)$ is defined in Equations (4.10) and (4.11).

---

[7] See [27] for more details.

$$\bar{\Omega}\left(\hat{\omega}_k^+\right) \equiv \begin{bmatrix} I_{3x3}\cos\left(\frac{1}{2}\left\|\hat{\omega}_k^+\right\|\Delta t\right) - \left[\hat{\Psi}_k^+ \times\right] & \hat{\Psi}_k^+ \\ -\hat{\Psi}_k^{+T} & \cos\left(\frac{1}{2}\left\|\hat{\omega}_k^+\right\|\Delta t\right) \end{bmatrix} \tag{4.10}$$

$$\hat{\Psi}_k^+ \equiv \frac{\sin\left(\frac{1}{2}\left\|\hat{\omega}_k^+\right\|\Delta t\right)\hat{\omega}_k^+}{\left\|\hat{\omega}_k^+\right\|} \tag{4.11}$$

The discrete sample time is denoted by Δt. The next propagation models to be defined are the angular velocity and the gyro bias. These models follow the standard EKF formulation given the post-update bias $\hat{\beta}_k^+$.

$$\hat{\omega}_k^+ = \tilde{\omega}_k - \hat{\beta}_k^+ \qquad \hat{\beta}_k^- = \hat{\beta}_k^+ \tag{4.12}$$

The error covariance propagation follows the same approach as in Chapter V. The discrete model is derived using a power series like was done for the quaternion update. This leads to Equation(4.13) where Φ_k, G_k, and Q_k are defined in Equations (4.14), (4.15), and (4.16).

$$P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + G_k Q_k G_k^T \tag{4.13}$$

$$\Phi_k = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix}$$

$$\Phi_{11} = I_{3x3} - \left[\hat{\omega}_k^+ \times\right]\frac{\sin\left(\left\|\hat{\omega}_k^+\right\|\Delta t\right)}{\left\|\hat{\omega}_k^+\right\|} + \left[\hat{\omega}_k^+ \times\right]^2 \frac{\left\{1-\cos\left(\left\|\hat{\omega}_k^+\right\|\Delta t\right)\right\}}{\left\|\hat{\omega}_k^+\right\|^2}$$

$$\Phi_{12} = \left[\hat{\omega}_k^+ \times\right]\frac{\left\{1-\cos\left(\left\|\hat{\omega}_k^+\right\|\Delta t\right)\right\}}{\left\|\hat{\omega}_k^+\right\|^2} - I_{3x3}\Delta t - \left[\hat{\omega}_k^+ \times\right]^2 \frac{\left\{\left\|\hat{\omega}_k^+\right\|\Delta t - \sin\left(\left\|\hat{\omega}_k^+\right\|\Delta t\right)\right\}}{\left\|\hat{\omega}_k^+\right\|^3} \tag{4.14}$$

$$\Phi_{21} = 0_{3x3}$$

$$\Phi_{22} = I_{3x3}$$

$$G_k = \begin{bmatrix} -I_{3x3} & 0_{3x3} \\ 0_{3x3} & I_{3x3} \end{bmatrix} \tag{4.15}$$

$$Q_k = \begin{bmatrix} \left(\sigma_v^2\Delta t + \frac{1}{3}\sigma_u^2\Delta t^3\right)I_{3x3} & -\left(\frac{1}{2}\sigma_u^2\Delta t^2\right)I_{3x3} \\ -\left(\frac{1}{2}\sigma_u^2\Delta t^2\right)I_{3x3} & \left(\sigma_u^2\Delta t\right)I_{3x3} \end{bmatrix} \tag{4.16}$$

Where σ_v^2 and σ_u^2 are the variances

Now the update equations need to be determined. To start, the estimated error state definition is $\Delta\hat{\hat{\mathbf{x}}}_k^+ \equiv \begin{bmatrix} \Delta\hat{\alpha}_k^{+T} & \Delta\hat{\beta}_k^{+T} \end{bmatrix}$. Using the small angle approximation $\delta\varsigma \approx \delta\alpha / \mathbf{2}$ and q_4≈1, the four state quaternion has been replaced by the three state Euler error angle vector. This minimizes use of the factors ½ and 2 in the EKF. It also allows the 3σ bounds to be directly calculated from the error covariance matrix.

The equations for the calculation of the observation matrix, H_k, for the different sensor types will now be derived. First, it must be determined what each sensor is actually measuring. The star tracker will provide a quaternion solution as its measurements, each sun sensor will provide a body vector referenced to the sun, and the magnetometer will also provide a body vector but referenced to the Earth. If the star tracker were to provide body vectors referenced to the stars, its update would be the same as the sun sensor update. This dose, however, require access to the star catalog and since the star tracker is being treated as a black box this access cannot be assumed. Next, the measurement model will be developed. The definition of the measurement is

$$\mathbf{y}_k \equiv \mathbf{h}_k\left(\hat{\mathbf{x}}_k\right) + \mathbf{v}_k \tag{4.17}$$

For the star tracker that provides a quaternion as the measurement, Equation(4.17) becomes

$$\mathbf{y}_k = \mathbf{q}_k + \mathbf{v}_k \tag{4.18}$$

The actual attitude quaternion, $\mathbf{q}_k$, is related to the propagated measurement, $\mathbf{q}_k^-$, through

$$\mathbf{q}_k = \delta\mathbf{q}_k \otimes \mathbf{q}_k^- = \begin{bmatrix} \Xi\left(\mathbf{q}_k^-\right) & \mathbf{q}_k^- \end{bmatrix} \delta\mathbf{q}_k \tag{4.19}$$

Using Equation(4.19) in the observation matrix definition Equation(4.20) leads to the observation matrix for the star tracker becoming

$$H_k\left(\hat{\mathbf{x}}_k^-\right) \equiv \left.\frac{\partial \mathbf{h}_k\left(\hat{\mathbf{x}}_k\right)}{\hat{\mathbf{x}}_k^-}\right|_{\hat{\mathbf{x}}_k^-} \tag{4.20}$$

$$ {}^{ST}H_k\left(\hat{\mathbf{x}}_k^-\right) = \begin{bmatrix} \frac{1}{2}\Xi\left(\mathbf{q}_k^-\right) & 0_{3x3} \end{bmatrix} \tag{4.21} $$

The sun sensor provides body vectors for which the true and estimated vectors are defined as

$$ \mathbf{b}_k \equiv A\left(\mathbf{q}_k\right)\mathbf{r} \quad \hat{\mathbf{b}}_k^- \equiv A\left(\hat{\mathbf{q}}_k^-\right)\mathbf{r} \tag{4.22} $$

Where the actual attitude matrix, $A\left(\mathbf{q}_k\right)$, and the estimated attitude matrix, $A\left(\hat{\mathbf{q}}_k^-\right)$, are related through

$$ A\left(\mathbf{q}_k\right) = A\left(\delta\mathbf{q}_k\right)A\left(\hat{\mathbf{q}}_k^-\right) \tag{4.23} $$

With the error-attitude matrix approximated with

$$ A\left(\delta\mathbf{q}_k\right) \approx I_{3x3} - \left[\alpha \times\right] \tag{4.24} $$

By substituting Equations (4.23) and (4.24) into Equation(4.22) and using the result in Equation(4.20) the observation matrix for the sun sensors becomes

$$ {}^{SS}H_k\left(\hat{\mathbf{x}}_k^-\right) = \begin{bmatrix} \left[A\left(\hat{\mathbf{q}}_k^-\right)\mathbf{r} \times\right] & 0_{3x3} \end{bmatrix} \tag{4.25} $$

Finally, the magnetometer also measures a body vector as defined in Equation(4.22). However, due of the highly nonlinear nature of the Earth's magnetic field and the resulting uncertainty in the estimated magnetic field it will be developed using the perturbation technique. Start with the equations for propagated and measured magnetic field respectively.

$$ B\left(\mathbf{x}_k^-\right) = D_b^I B_I + \Delta D_b^I B_I + \nu_p \qquad B_m = D_b^I B_I + \nu_m \tag{4.26} $$

Where D_b^I is the transformation from inertial to body coordinates, $\Delta D_b^I B_I$ is the estimation error, and ν_p and ν_m are the errors in the magnetic field model and measurement respectively. Defining $\nu \equiv \nu_m - \nu_p$, the residual becomes

$$ z = -\Delta D_b^I B_I + \nu \tag{4.27} $$

The transformation error, ΔD_b^I, can be defined as the difference between the true inertial to body transform and the estimated inertial to body transform. The estimated transform can be written $D_c^b D_b^I$. Where D_c^b is the transform from the body to computed coordinates. Now, assuming D_c^b is made of small angles, the transformation error can be rewritten as

$$\Delta D_b^I = \left(I - \left[\Delta\hat{\alpha}_k^+ \times\right]\right) D_b^I - D_b^I = -\left[\Delta\hat{\alpha}_k^+ \times\right] D_b^I \tag{4.28}$$

Where $\left[\Delta\hat{\alpha}_k^+ \times\right]$is the cross product matrix of Euler error angles. Substituting Equation(4.28) into Equation(4.27) leads to

$$z = \left[\left[-B_b \times\right] \quad 0_{3x3}\right] \Delta\hat{\mathbf{x}}_k^+ + \nu \tag{4.29}$$

The observation matrix follows in Equation(4.30), since B_b is unknown B_m is substituted.

$${}^B H_k = \left[\left[-B_m \times\right] \quad 0_{3x3}\right] \tag{4.30}$$

Finally, the remaining update equations need to be determined. The standard EKF form for the error state vector update is .

$$\Delta\hat{\mathbf{x}}_k^+ = K_k \left[\mathbf{y}_k - \mathbf{h}_k\left(\hat{\mathbf{x}}_k^-\right)\right] \tag{4.31}$$

Where, $\mathbf{y}_k$ is the measurement and $\mathbf{h}_k\left(\hat{\mathbf{x}}_k^-\right)$is the estimated measurement. For the star tracker the estimated measurement is the propagated quaternion, $\hat{\mathbf{q}}_k^-$. For a body vector it has a form $\mathbf{h}_k\left(\hat{\mathbf{x}}_k^-\right) = A\left(\hat{\mathbf{q}}_k^-\right)\mathbf{r}$, where $\mathbf{r}$ is the propagated vector to the reference body, i.e. the Sun. The bias update is

$$\hat{\beta}_k^+ = \hat{\beta}_k^- + \Delta\hat{\beta}_k^+ \tag{4.32}$$

The quaternion update must us quaternion multiplication like the quaternion error did in Equation(4.2). The quaternion update equation is

$$\hat{\mathbf{q}}_k^+ = \hat{\mathbf{q}}_k^- + \frac{\Xi\left(\hat{\mathbf{q}}_k^-\right)}{2} \delta\hat{\alpha}_k^+ \tag{4.33}$$

The error covariance matrix update and the gain calculation, both have the same form from Chapter V.

$$P_k^+ = \left[I - K_k H_k\left(\hat{\mathbf{x}}_k^-\right)\right] P_k^- \tag{4.34}$$

$$K_k = P_k^- H_k^T\left(\hat{\mathbf{x}}_k^-\right)\left[H_k\left(\hat{\mathbf{x}}_k^-\right) P_k^- H_k^T\left(\hat{\mathbf{x}}_k^-\right) + R_k\right]^{-1} \tag{4.35}$$

The Discrete Multiplicative Quaternion Extended Kalman Filter equations are summarized in Table 13.

Discrete Multiplicative Quaternion Extended Kalman Filter	
Initialize	$\hat{\mathbf{q}}(k_0) = \hat{\mathbf{q}}_0, \quad \beta(k_0) = \beta_0$ $P(k_0) = P_0$
Propagate	$\hat{\omega}_k^+ = \tilde{\omega}_k - \hat{\beta}_k^+$ $\hat{\beta}_k^- = \hat{\beta}_k^+$ $\hat{\mathbf{q}}_{k+1}^- = \bar{\Omega}\left(\hat{\omega}_k^+\right)\hat{\mathbf{q}}_k^+$ $P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + G_k Q_k G_k^T$
Gain	$K_k = P_k^- H_k^T\left(\hat{\mathbf{x}}_k^-\right)\left[H_k\left(\hat{\mathbf{x}}_k^-\right)P_k^- H_k^T\left(\hat{\mathbf{x}}_k^-\right) + R_k\right]^{-1}$ ${}^{ST}H_k\left(\hat{\mathbf{x}}_k^-\right) = \left[\frac{1}{2}\Xi\left(\mathbf{q}_k^-\right) \quad 0_{3x3}\right]$ ${}^{SS}H_k\left(\hat{\mathbf{x}}_k^-\right) = \left[\left[A\left(\hat{\mathbf{q}}_k^-\right)\mathbf{r}\times\right] \quad 0_{3x3}\right]$ ${}^{B}H_k = \left[\left[-B_m\times\right] \quad 0_{3x3}\right]$
Update	$\Delta\hat{\tilde{\mathbf{x}}}_k^+ \equiv \left[\Delta\hat{\alpha}_k^{+T} \quad \Delta\hat{\beta}_k^{+T}\right]$ $\Delta\hat{\tilde{\mathbf{x}}}_k^+ = K_k\left[\mathbf{y}_k - \mathbf{h}_k\left(\hat{\mathbf{x}}_k^-\right)\right]$ $\hat{\beta}_k^+ = \hat{\beta}_k^- + \Delta\hat{\beta}_k^+$ $\hat{\mathbf{q}}_k^+ = \hat{\mathbf{q}}_k^- + \frac{\Xi\left(\hat{\mathbf{q}}_k^-\right)}{2}\delta\hat{\alpha}_k^+$ $P_k^+ = \left[I - K_k H_k\left(\hat{\mathbf{x}}_k^-\right)\right]P_k^-$

Table 13. Discrete Multiplicative Quaternion Extended Kalman Filter. After [27].

B. IMPLEMENTATION

The EKF described in Table 13 is incomplete without some of the particular implementation items that must be addressed. The initialization, expected measurement noise, and quaternion normalization are a few. all be covered to help complete the full development of this EKF.

1. Initialization

In an EKF initialization can be very crucial to the performance. Here the initial state was chosen to be $\hat{\mathbf{x}}_0^- = \begin{bmatrix} \mathbf{q}_0 & 0_{3x1} \end{bmatrix}$, where $\mathbf{q}_0 = \begin{bmatrix} 0_{3x1} & 1 \end{bmatrix}$. The error covariance matrix is a little more involved. Since all of the noise sources in the sensors and propagation model have estimated variances (from Chapter III), an estimation can be calculated for the error covariance. Using Farrenkopf's steady-state analysis from section 7.2.4 in [27], one can calculate a rough estimate to the error covariance. The following equation was used to estimate P_0.

$$P_0 = \Delta t^{1/4} \sigma_n^{1/2} \left(\sigma_v^2 + 2\sigma_u \sigma_v \Delta t^{1/2} \right)^{1/4} \tag{4.36}$$

Where $\sigma_n = \sqrt{\sigma_{ST}^2 + \sigma_{SS}^2 + \sigma_{Mag}^2}$, σ_v^2 is the variance associated with η_v , and σ_u^2 is the variance associated with η_u both from Equation(2.15).

2. Measurement Noise

The expected measurement noise was calculated for each sensor in Chapter III. They are used in the R matrix when the Kalman gain is calculated. Each estimated measurement noise matrix follows the form

$$R_k = \sigma^2 I_{3x3} \tag{4.37}$$

The actual values of noise variance used are summarized in Table 14.

Sensor	Value	Units
Star tracker	$\sigma_{st}^2 = 1.2797E-8$	rad
Sun Sensor	$\sigma_{ss}^2 = 3.0462E-6$	rad
Magnetometer	$\sigma_{mag}^2 = 1.25E-6$	tesla

Table 14. Measurement Noise Variance Values.

3. Quaternion Normalization

The quaternion update used in the formulation of the EKF, Equation(4.33), is only guaranteed to be a unit vector to within the first-order. This means normalization needs to occur to prevent the build up of computational errors. Two methods were used to help maintain a normalized quaternion. The first method used does not formally normalize the quaternion, but reduces the error to order $\varepsilon^3 / 32$. This factor is suggested in [29], computed in Equation(4.38), and is multiplied by the quaternion update when $\hat{\mathbf{q}}_k^{-T}\hat{\mathbf{q}}_k^-$ falls between two predetermined error bounds.

$$\hat{\mathbf{q}}_k^- = \left(\frac{3+\hat{\mathbf{q}}_k^{-T}\hat{\mathbf{q}}_k^-}{1+3\hat{\mathbf{q}}_k^{-T}\hat{\mathbf{q}}_k^-}\right)\hat{\mathbf{q}}_k^- \qquad (4.38)$$

The second method is the brut force normalization per Equation(4.39). The full normalization is only done when $\hat{\mathbf{q}}_k^{-T}\hat{\mathbf{q}}_k^-$ becomes greater than the upper error bound. This reduces the computational burden for on-board processing because it is simple a scalar multiplied by a vector the majority of the time. How often this is done should be determined through simulation to establish an acceptable error tolerance for the application.

$$\hat{\mathbf{q}}_k^- = \frac{\hat{\mathbf{q}}_k^-}{\sqrt{\hat{\mathbf{q}}_k^{-T}\hat{\mathbf{q}}_k^-}} \qquad (4.39)$$

4. Murrell's Version

The application of superposition to the EKF was discussed in Chapter IV. To actually implement it in the EKF some structural changes were made from the standard form. The most significant change, however, is in the estimated error state update in Equation(4.31). The update now must include any previous estimates of the error state from that set of measurements and only add error unaccounted for in these previous estimates. Equation(4.40) replaces Equation(4.31) in the above EKF.

$$\Delta\hat{\tilde{\mathbf{x}}}_k^+ = \Delta\hat{\tilde{\mathbf{x}}}_k^- + K_k\left[\mathbf{y}_k - \mathbf{h}_k\left(\hat{\mathbf{x}}_k^-\right) - H_k\Delta\hat{\tilde{\mathbf{x}}}_k^-\right] \tag{4.40}$$

VII. SIMULATION RESULTS

The simulations performed here are only preliminary. They have not been verified through a Monte Carlo simulation or hardware-in-the-loop simulations. No disturbance torques are applied during any simulation. No control torque is applied during any of the simulations, as only the error between the true value and the estimated value is needed. This means the no particular attitude is actively maintained, however, the satellite will be initialized nadir pointing and will rotate at orbital velocity. This should maintain a nadir pointing attitude because there are no disturbances. During each simulation selected sensors will be used in the EKF to estimate the attitude. The estimate is compared to the actual attitude in Figure 24 and Figure 25 to form the errors. Table 15 shows the simulation conditions for each run. Additional Figures of simulation results are in the Appendix.

Simulation	Sensors[8] (ST, SS1, SS2, Mag)	Initial ω[9] (inertial)	Initial Attitude (inertial)	EKF[10] (Hz)	Sensor[11] (Hz)
1	(1,0,0,0)	(0,2π/P,0)	(0,0,0)	20	1
2	(1,1,1,0)	(0,2π/P,0)	(0,0,0)	20	1,5,5
3	(0,1,1,1)	(0,2π/P,0)	(0,0,0)	20	5,5,20
4	(1,1,1,1)	(0,2π/P,0)	(0,0,0)	20	1,5,5,20

Table 15. Simulation Conditions.

---

[8] 1= On, 0 =Off

[9] 2π/P is the orbital velocity wrt the inertial frame.

[10] The simulation and the EKF ran at the same speed.

[11] The sample rates of each selected sensor is shown in the order of column 1.

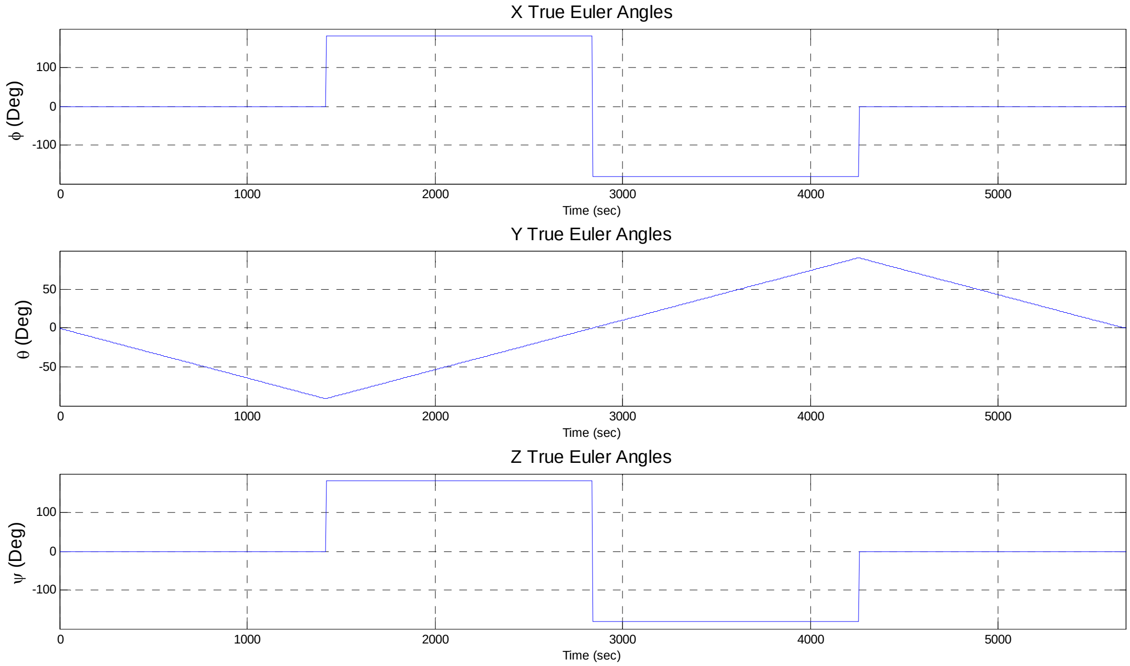


Figure 24. True Euler Angles in Inertial Frame[12].

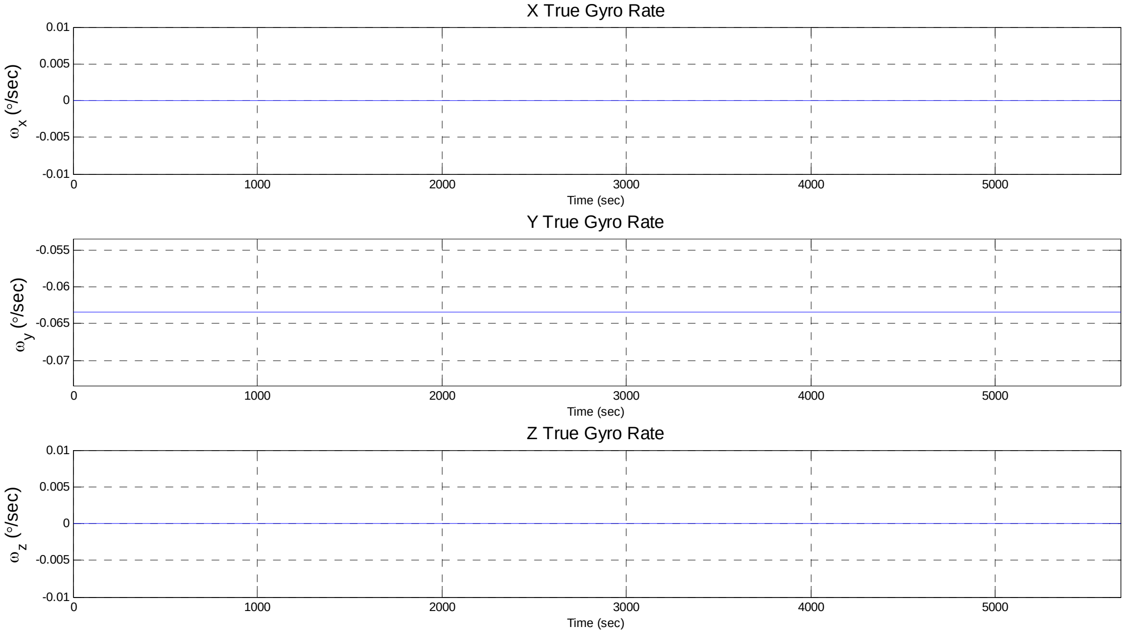


Figure 25. True Angular Rates in Inertial Frame.

---

[12] The jumps in angle is due to the range limitations of the sine and tan2 functions.

A. EKF PERFORMANCE WITH NOISY STAR TRACKER

During this simulation, the EKF will only use the quaternion measurements from the star tracker. This will set a baseline for other configurations to compare against. The expected Euler angle error is less than 0.026 ° 1σ. The rms value of the error will be calculated and used for comparison. The 3σ boundaries (red) are directly calculated from the error covariance matrix. Figure 27 and Figure 28 show that the Euler error and the bias error are in fact well bound. They also show that the rms error is within the expected limits. An unexpected result is also apparent. A spike in the error covariance, along with degradation in the Euler error at ~2853 sec occurs. It has been determined the cause of this error is due to the scalar quaternion equaling zero at this point (see Figure 30). This causes the gain to become very large and in turn cause the increase in *P*. Figure 29 and Figure 31 show a weakness of the EKF, it does not filter the gyro noise well because it is used in the propagation process and not as a measurement. This accounts for the minimal improvement in the Euler angle estimate over the star tracker expected error.

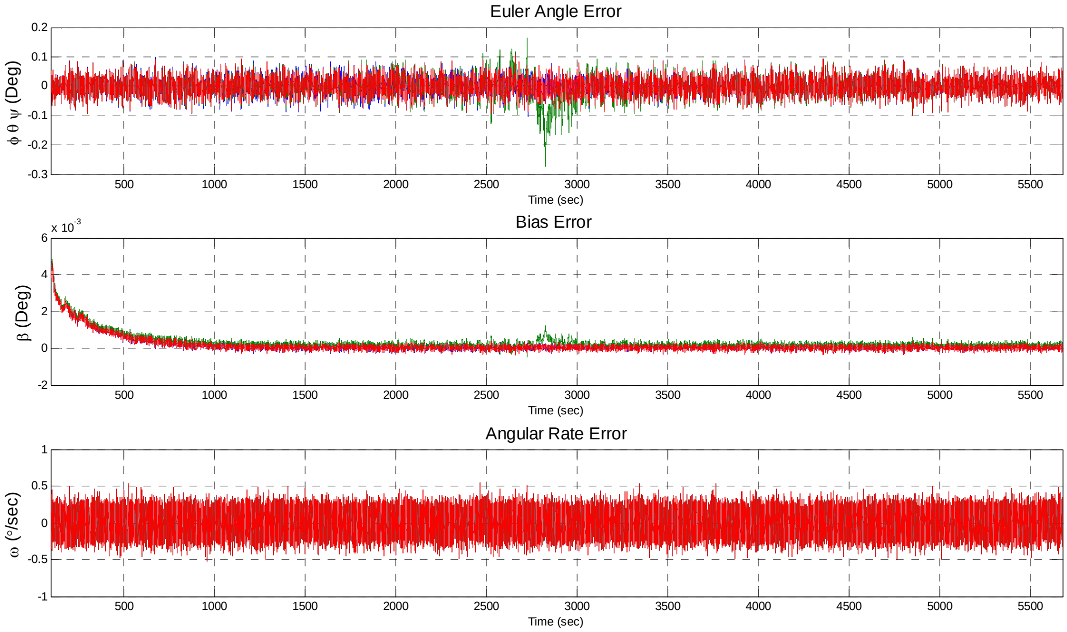


Figure 26. Simulation 1 Euler, Bias, and Angular Rate Error.

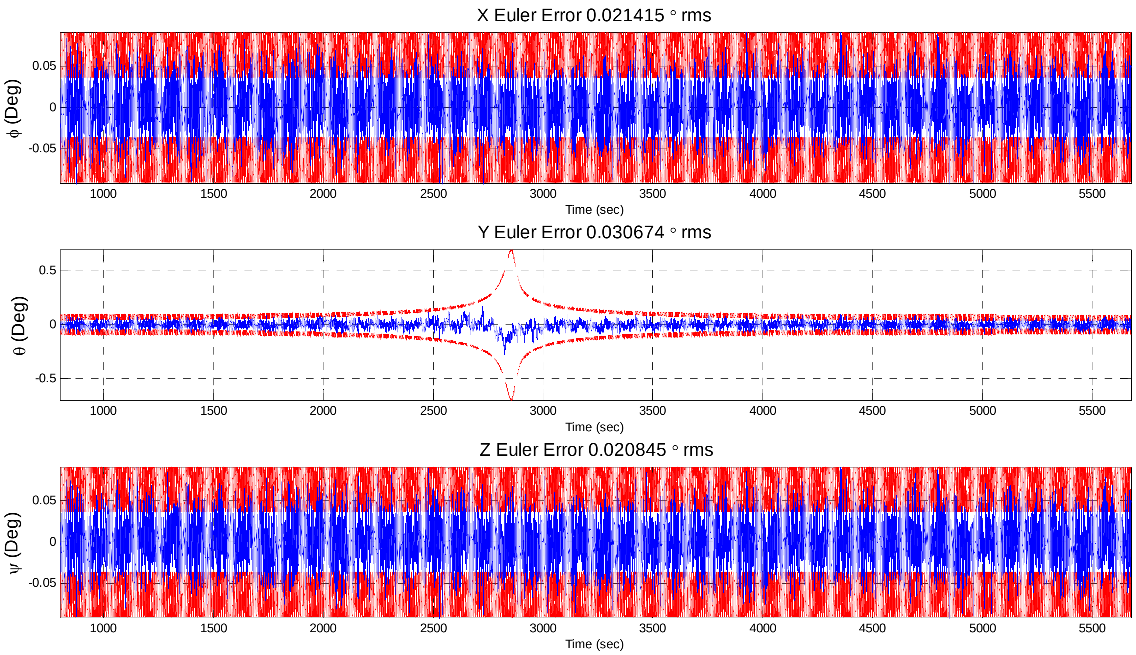


Figure 27. Simulation 1 Euler Angle Error with 3σ Boundaries.

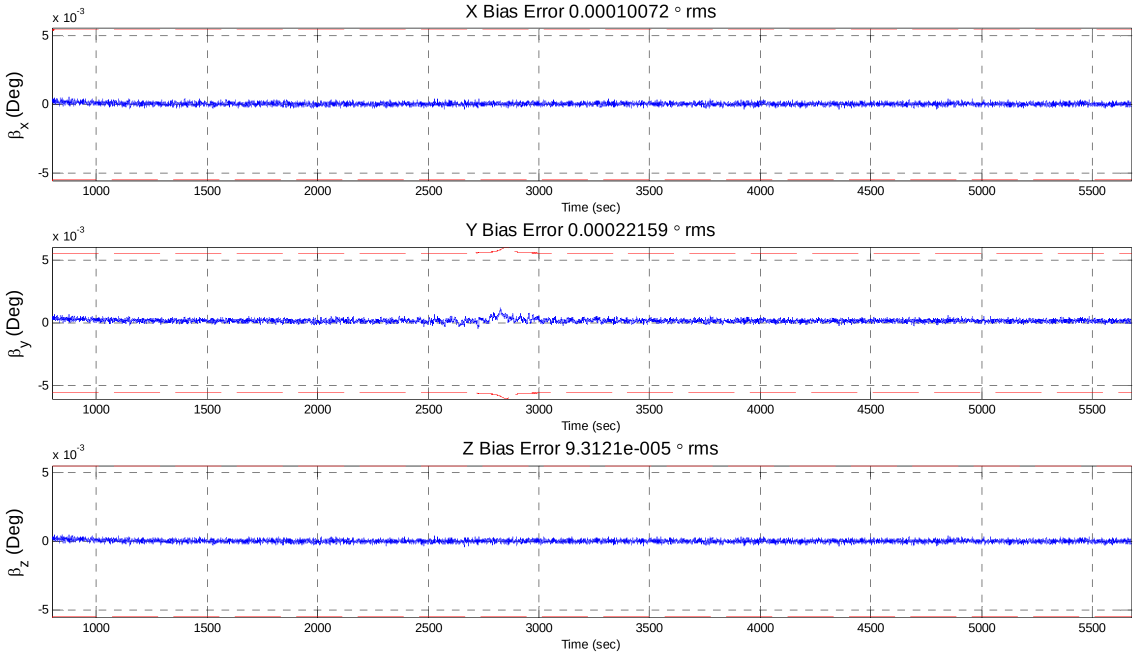


Figure 28. Simulation 1 Gyro Bias Error with 3σ Boundaries.

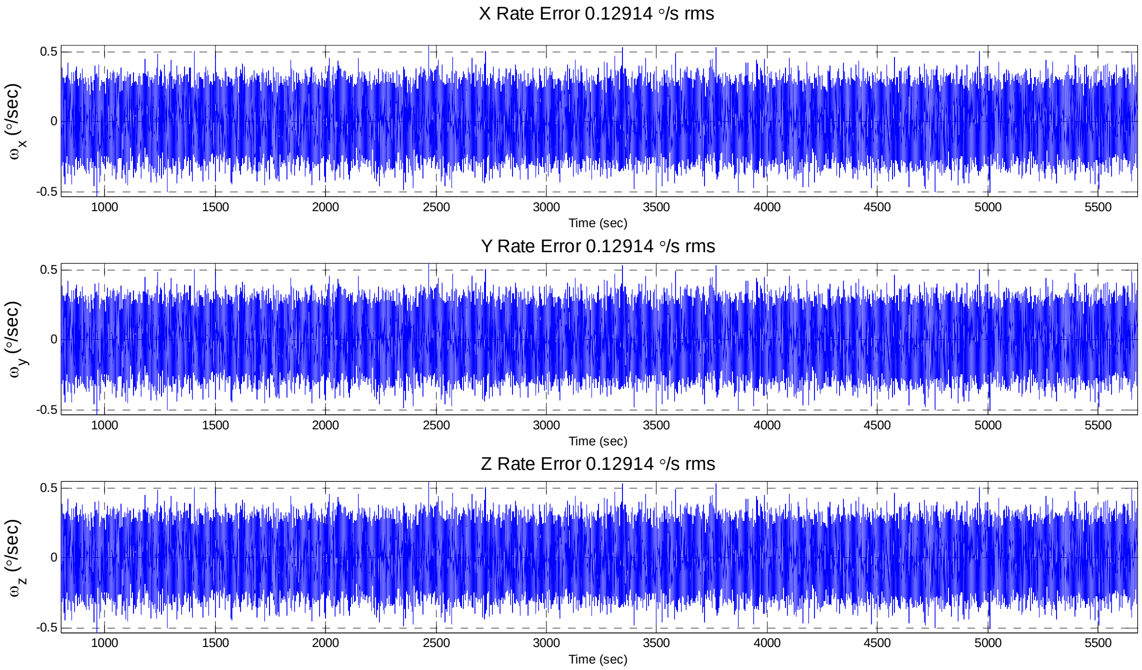


Figure 29. Simulation 1 Gyro Rate Error.

Figure 30 shows how the Gaussian noise in the Euler angles is no longer Gaussian in the quaternion. Figure 31 can be used for comparison with the rate error in Figure 29.

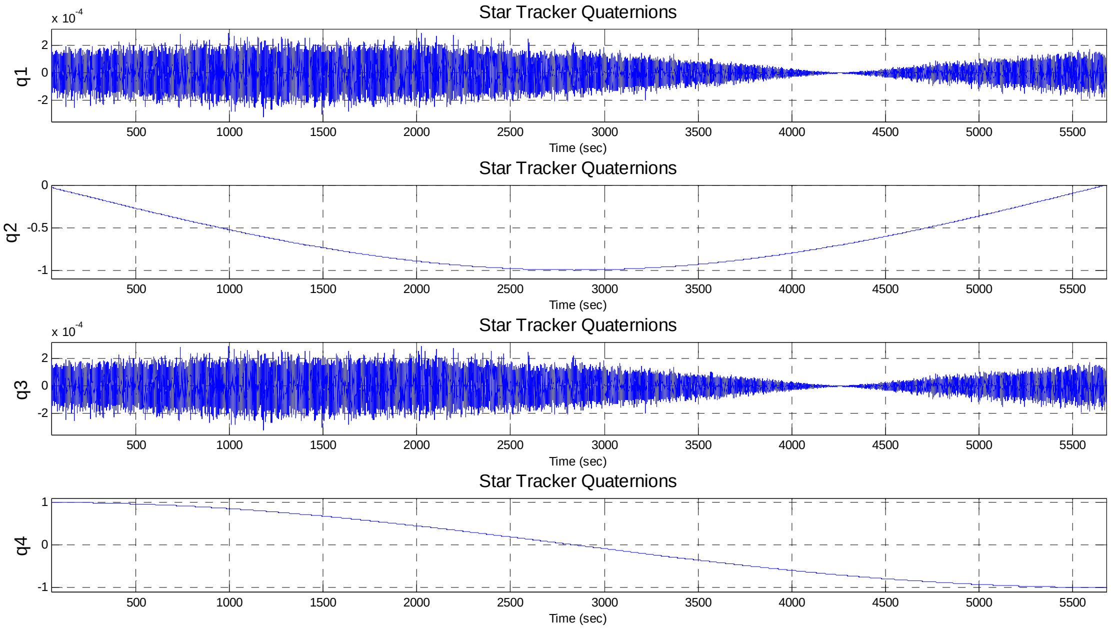


Figure 30. Simulation 1 Star Tracker Quaternion Measurements.

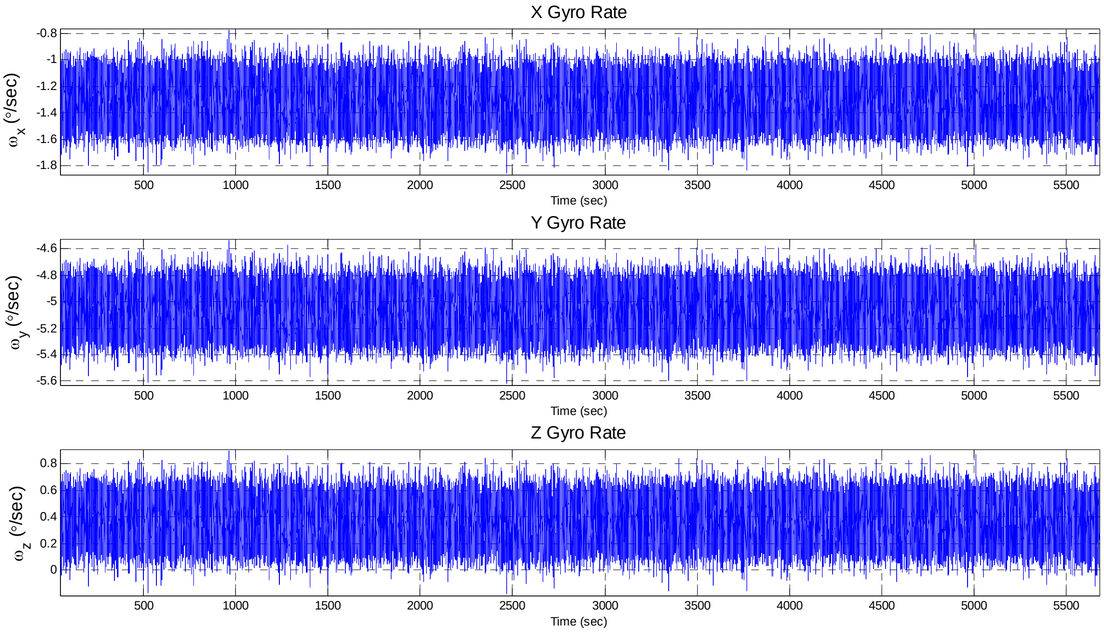


Figure 31. Simulation 1 Gyro Rate Measurements.

B. EKF PERFORMANCE WITH STAR TRACKER AND SUN SENSORS

This second simulation makes the sun sensor measurements available to the EKF as well as the star tracker quaternion. It is expected that an additional measurement will improve the estimate. Again the Euler angle error and the bias error are well bound by the 3σ boundaries in Figure 33 and Figure 34. The gyro bias rms error has improved from simulation 1. The lack of improvement is due to the anomaly discussed earlier. The sun sensors are unable to correct for this error because the satellite is in eclipse at the time. This can be seen in Figure 36 and Figure 37, i.e., at ~1800 seconds all measurements go to zero.

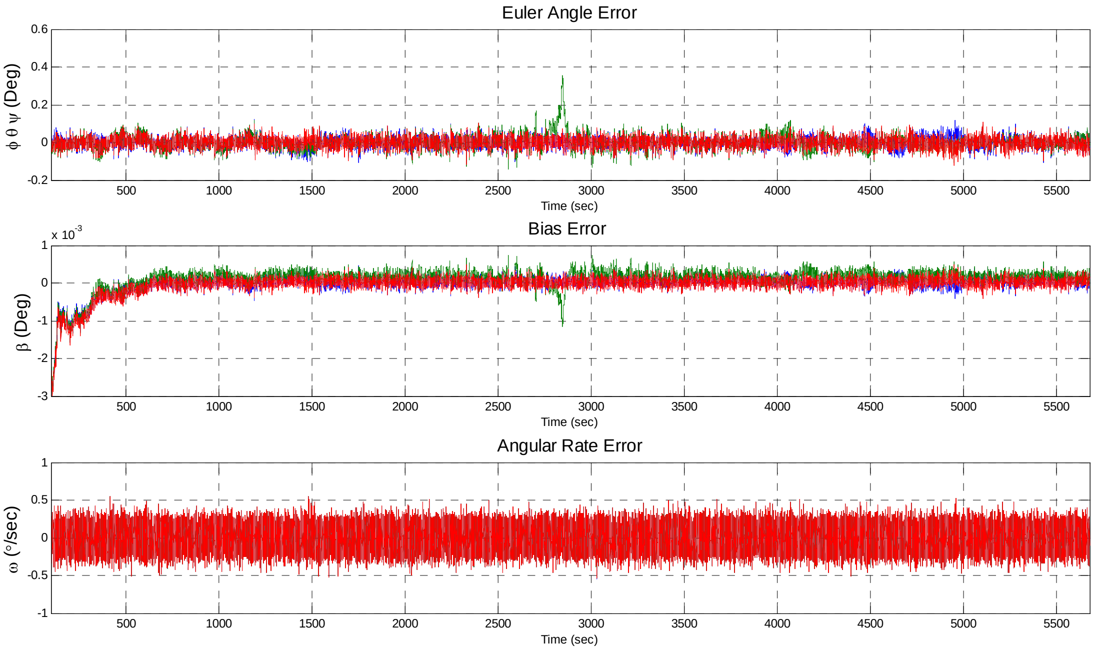


Figure 32. Simulation 2 Euler, Bias, and Angular Rate Error.

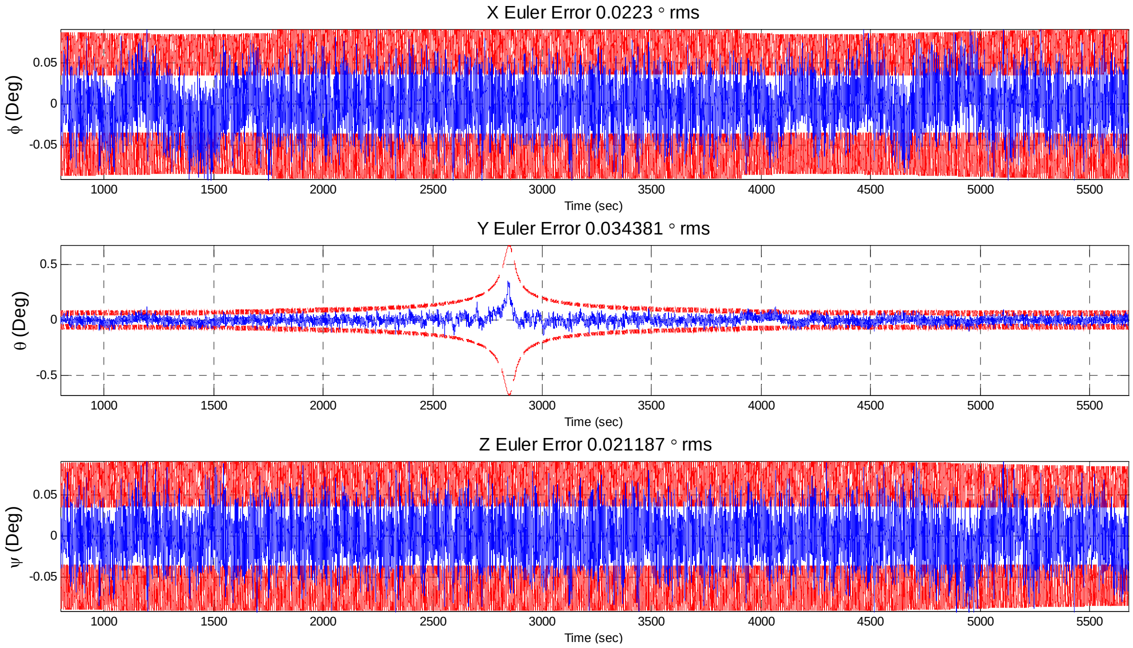
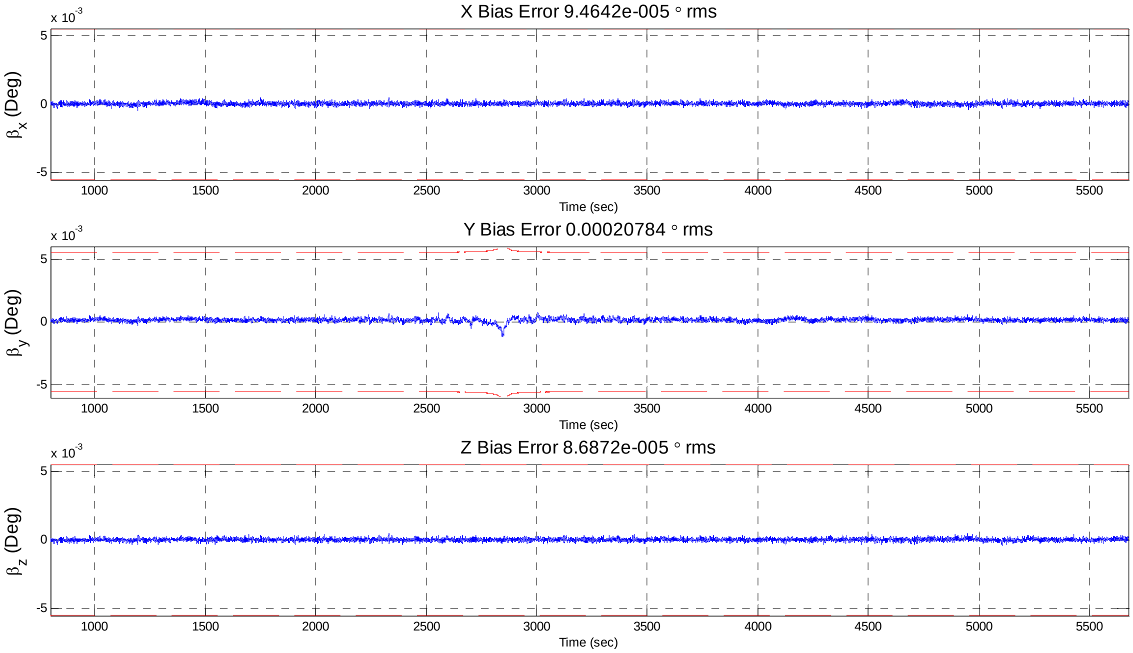


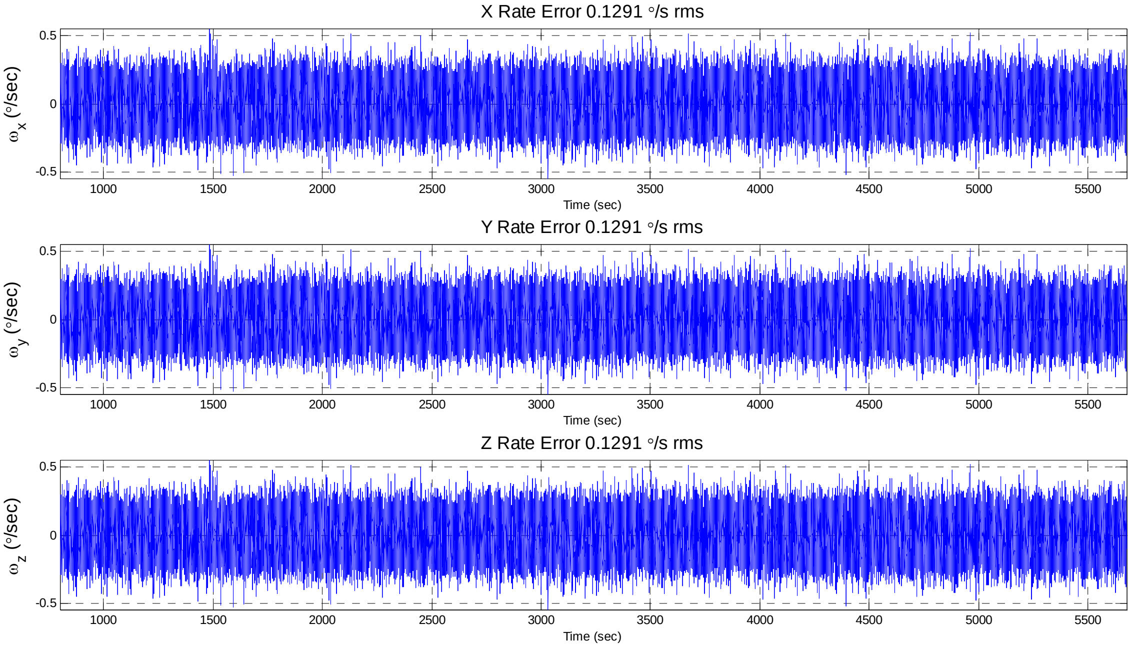
Figure 33. Simulation 2 Euler Error with 3σ Boundaries.


X Bias Error 9.4642e-005 ° rms

Y Bias Error 0.00020784 ° rms

Z Bias Error 8.6872e-005 ° rms

Figure 34. Simulation 2 Gyro Bias Error with 3σ Boundaries.


X Rate Error 0.1291 °/s rms

Y Rate Error 0.1291 °/s rms

Z Rate Error 0.1291 °/s rms

Figure 35. Simulation 2 Gyro Rate Error.

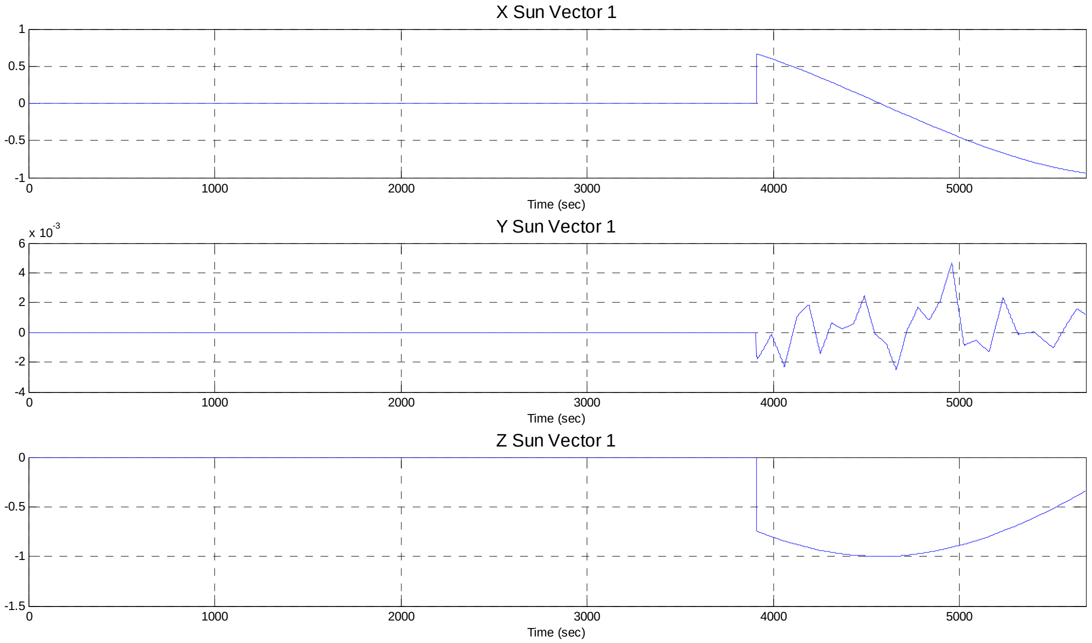


Figure 36. Simulation 2 Sun Sensor #1 Unit Vector Measurement.

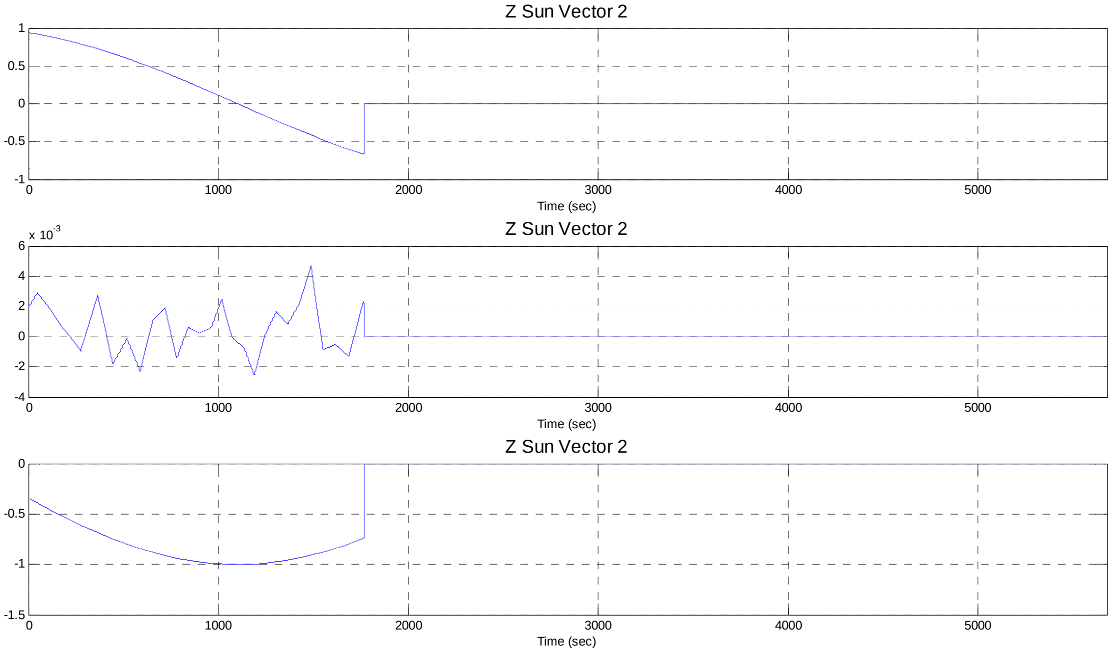


Figure 37. Simulation 2 Sun Sensor #2 Unit Vector Measurement.

C. EKF PERFORMANCE WITH SUN SENSORS AND MAGNETOMETER

During this simulation the sun sensor measurements and the magnetometer measurements will be used by the EKF to produce the estimate. The star tracker quaternion will not be used. It is expected that this configuration will take longer to reach steady state and be less accurate than the star tracker. Figure 39 and Figure 40 show that the rms errors are significantly less accurate than Simulation 1 or 2. What is interesting is that this simulation was more accurate than both ADCS "in a box" discussed in Chapter I. It is clear where the eclipse occurs on all the Figures due to the increase in errors, however, one should note that an rms error of approximately 1 ° rms is still maintained with only the magnetometer measurements.

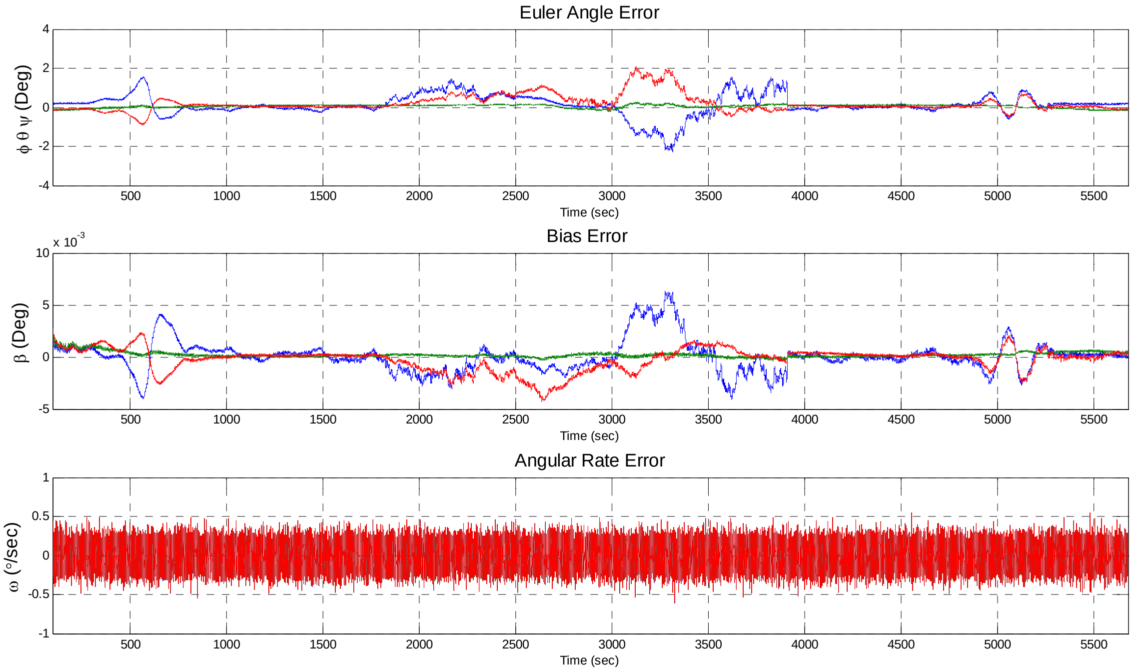


Figure 38. Simulation 3 Euler, Bias, and Angular Rate Error.

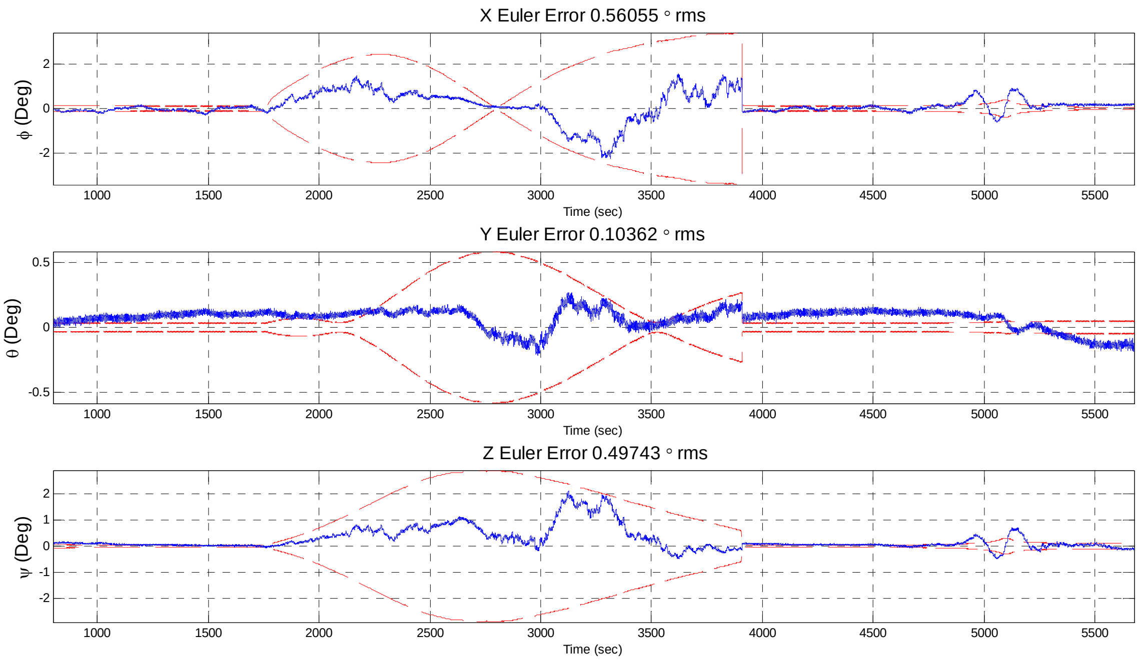


Figure 39. Simulation 3 Euler Error with 3σ Boundaries.

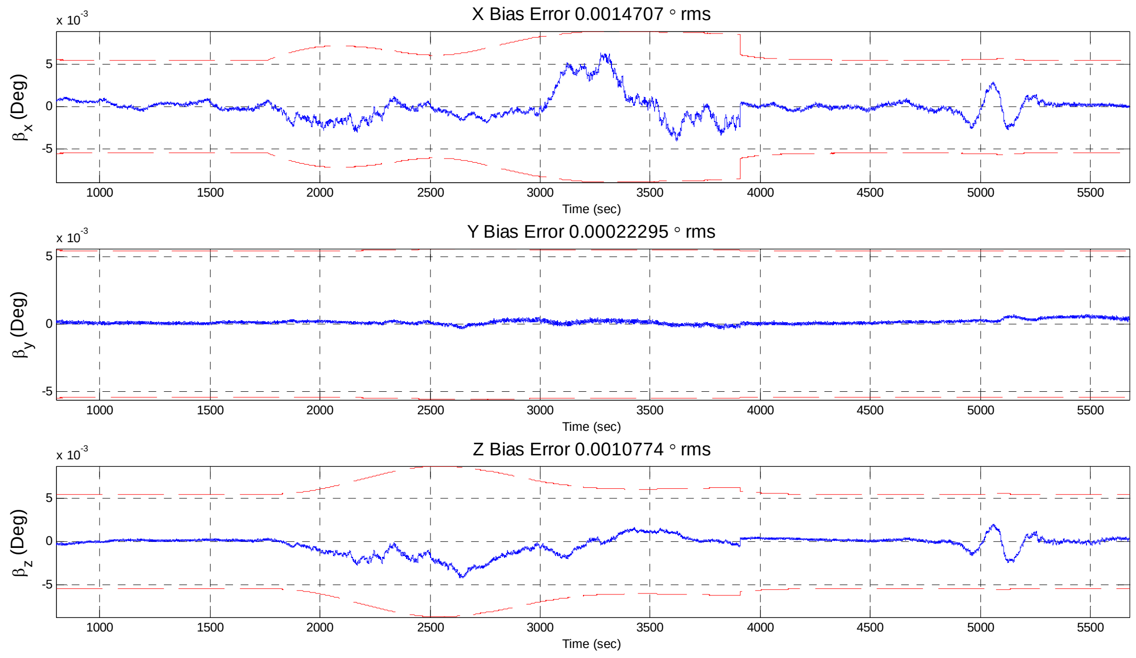


Figure 40. Simulation 3 Gyro Bias Error with 3σ Boundaries.

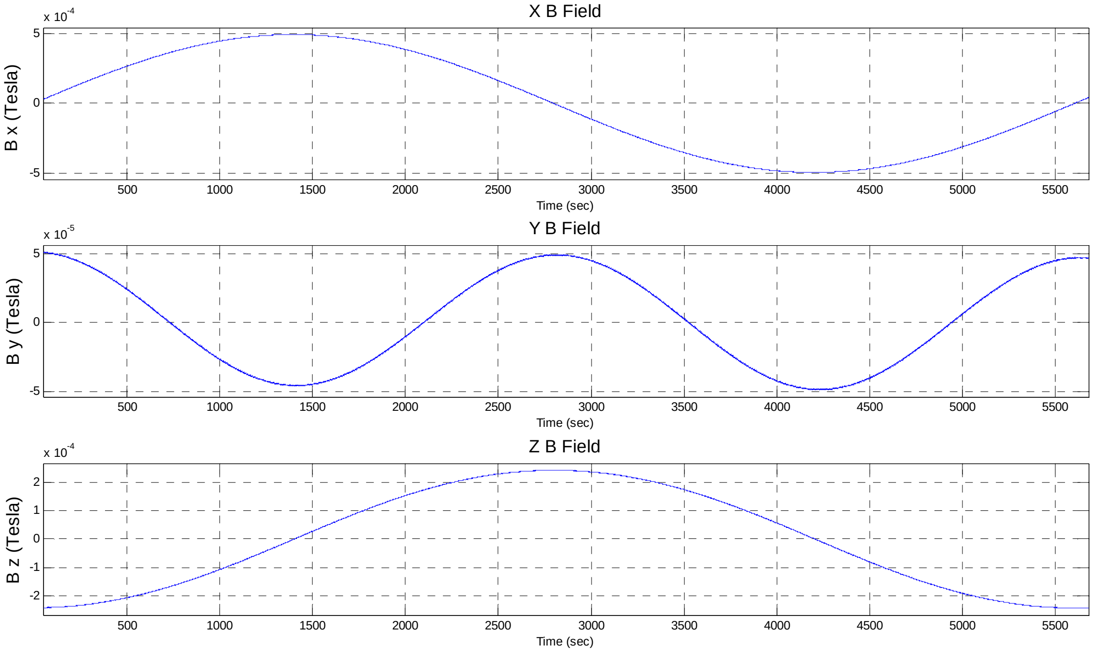


Figure 41. Simulation 3 Magnetometer Measurement.

Note on Figure 41 that the range has been changed to ±5 mGauss (5×10^{-4} Tesla) for the simulated magnetometer. The hardware range of ±3.5 mGauss proved too small and saturation occurred. This caused large errors in the state estimation. It was determined that the simulation should be run simulating a similar magnetometer with a larger dynamic range.

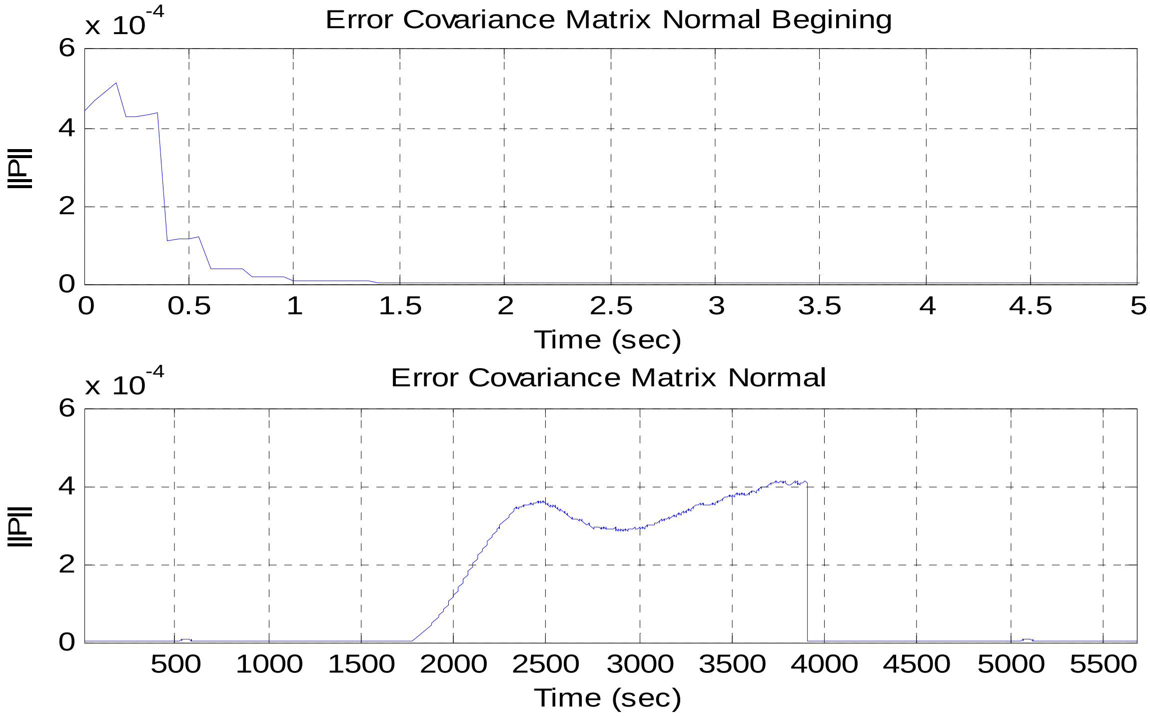


Figure 42. Simulation 3 Error Covariance Matrix Normal.

D. EFK PERFORMANCE WITH ALL SENSORS

Now all of the available sensor measurements are used in the EKF. It is expected that this configuration will produce the best estimate. It is obvious, however, that this is not the case. The Euler angle error is twice that of simulation 1. Figure 44 shows that the Euler error is not very well bound by the calculated 3σ boundaries. Figure 46 shows that the star tracker quaternion anomaly is still present, but no longer dominant. It appears that the magnetometer error is dominating the other sensors. The eclipse can also be clearly seen from 1800 to 3900 sec. The relatively small error covariance values suggest a noise value is not being accurately estimated. This is most likely from the magnetometer. It has the highest sample rate and the most error in attitude estimation. More work needs to be done on both the Earth's magnetic field model and the magnetometer modeling to improve the performance of the EKF.

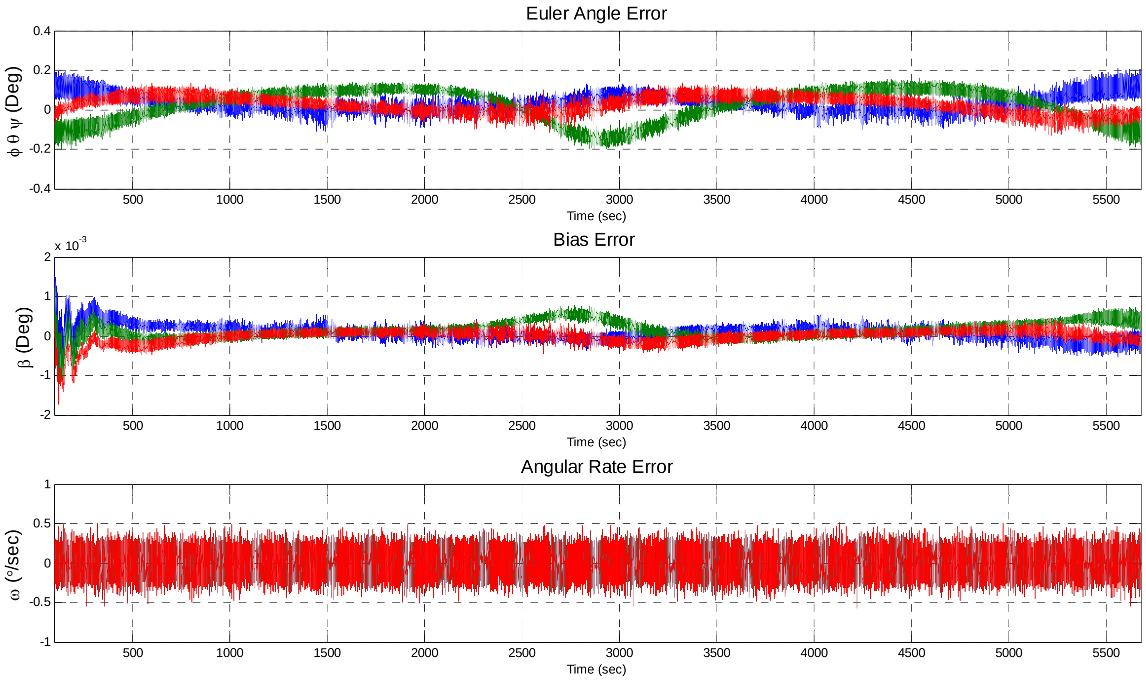


Figure 43. Simulation 4 Euler, Bias, and Angular Rate Error.

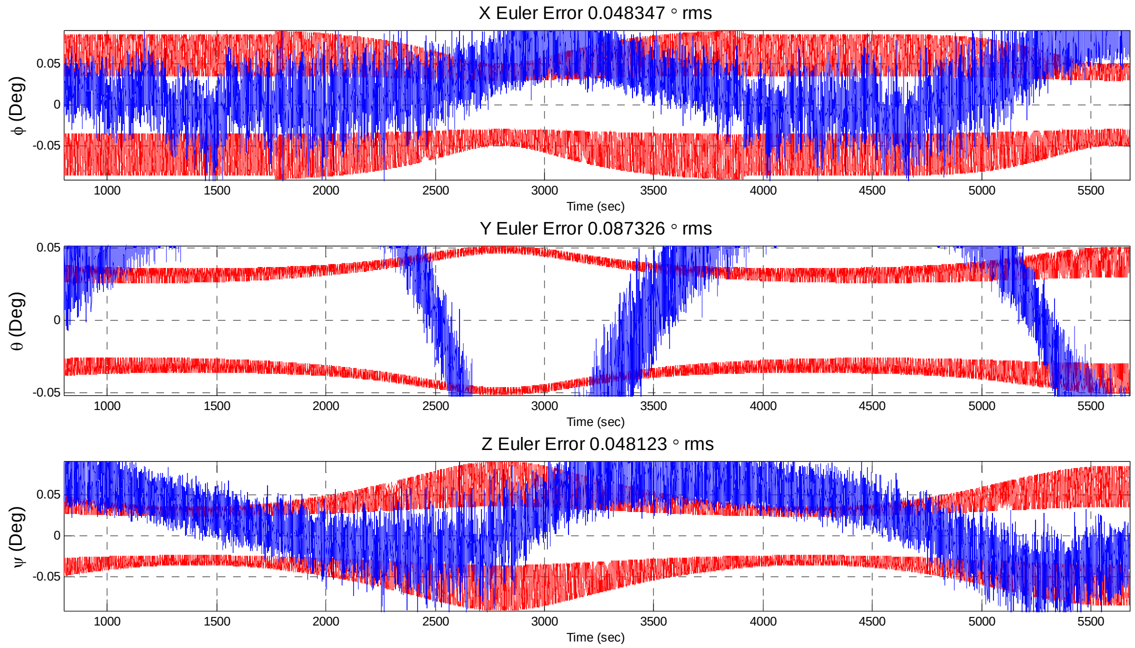


Figure 44. Simulation 4 Euler Error with 3σ Boundaries.

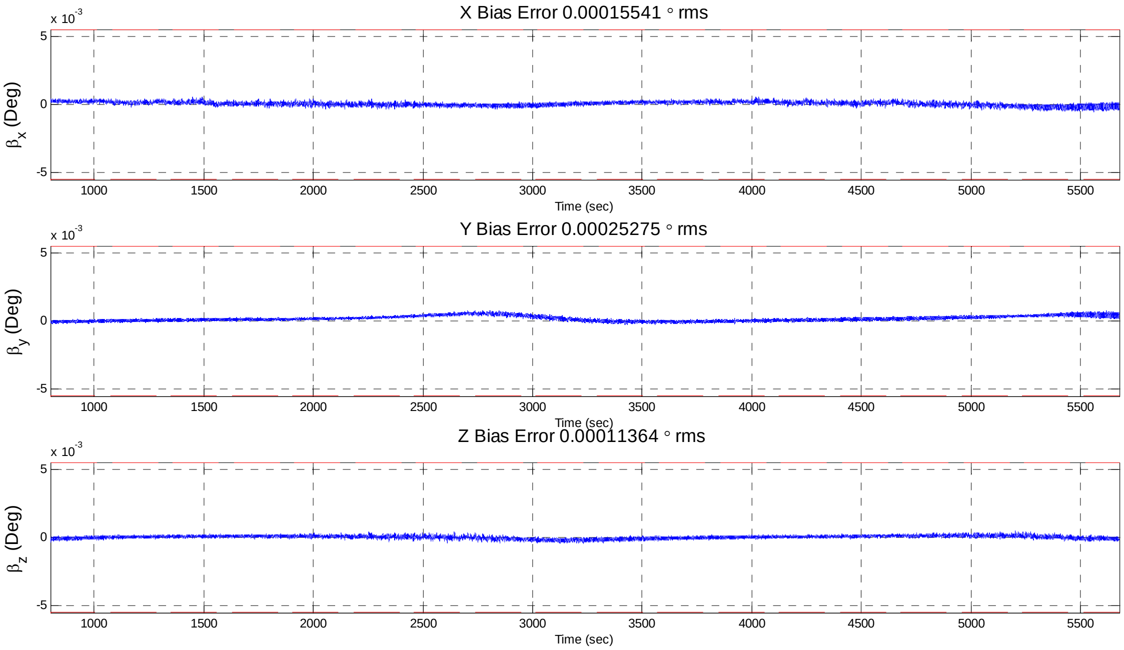


Figure 45. Simulation 3 Gyro Bias Error with 3σ Boundaries.

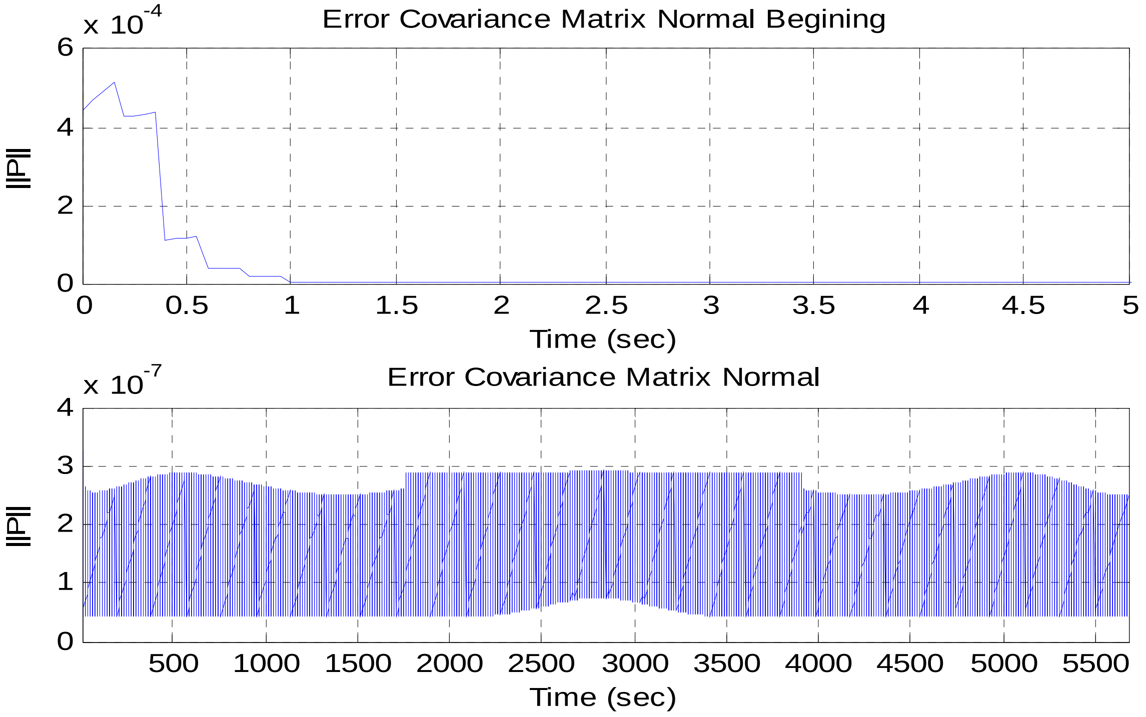


Figure 46. Simulation 4 Error Covariance Matrix Normal.

THIS PAGE INTENTIONALLY LEFT BLANK

VIII. CONCLUSION

A. SUMMARY

The results from the simulations clearly show that the EKF developed requires more tuning to achieve the expected performance. It has demonstrated utility in fusing the various sensors, but has not yet shown a great improvement in performance over the star tracker alone.

The anomaly, due to the star tracker quaternion causing the error covariance to dramatically increase, was an interesting result. A solution to this problem would be to use the centroids and star catalog of the star tracker like is done in [27]. This would avoid the problem encountered when the scalar quaternion equals zero. This sets of a chain of events which results in the error covariance increasing as well as actual reduced accuracy of the estimate. This only seems to occur because the scalar quaternion makes up the diagonal component of $\Xi\left(\hat{\mathbf{q}}_k^-\right)$. That makes the diagonal component of the H_k matrix zero, which is then used in the calculation of the gain. The gain then affects both the estimated state update and the error covariance update.

The solution to this problem requires the ADS have full access to all of the star tracker calculations and databases. This would essentially integrate the star tracker into the ADCS computer. An integrated ADS could then consist of star trackers, sun sensors, and magnetometers, as well as other sensors like horizon sensors, all connected and run by one processor to produce an attitude estimate.

The gyro model has been a great success, however, for the ADIS16405 hardware, modeling the bias instability may be more appropriate then the RRW. Modeling the bias instability would accurately account for the variations in the bias under all conditions. As was seen in Chapter II, the RRW is not a dominating noise in this gyro and may even be able to be filtered out. The bias instability, however, remains even with maximum filtering applied.

The magnetometer included in the ADIS16405 proved to have too small of a range for this application. The placement of the magnetometer is also problematic. The IMU is normally situated inside the spacecraft and given the small size of a CubeSat; it would consequently be close to magnetic interference from batteries and other electronics. Therefore, it is recommended to use a separate, external magnetometer in the final design of this ADS.

The general Simulink® simulation model has been significantly improved with the generalization done in the orbit propagation. The addition of the Earth's magnetic field and the improvement of the atmospheric density both provide a higher fidelity model of the space environment.

This thesis has not achieved all that was set out to do, but it has accomplished much. It has proven that data fusion is possible using Murrel's technique. It has also set ground work for the continued improvement and development of the TINYSCOPE Simulation. Most importantly, it has shown that a CubeSat can indeed have high accuracy attitude determination using currently existing technology.

B. FUTURE WORK

1. Verification and Testing

This thesis only presents preliminary results. A full Monte Carlo verification of the EKF performance should be performed to provide detailed performance parameters for the EKF and the simulation. Further verification of the EKF design can be performed with a 3-Axis Simulator. Running the EKF in a hardware-in-the-loop simulation will provide more realistic performance information on the EKF.

2. Further Develop Simulation

The sensor models developed for this thesis went a long way to providing realistic measurements for the ADCS. Further refinements to the Simulink® model can be made however. The star tracker model is still a fairly simple model and can be improved in many ways. Different sensors should also be developed such as a horizon sensor. The Earth's magnetic field certainly needs to have a more accurate model developed. A high order IGRF model should be developed that will provide a more realistic approximation of the actual magnetic field and can still run efficiently in the simulation.

3. Hardware

The development of new low cost, low power, miniature attitude sensors is accelerating. Continued evaluation of these new sensors should be performed as well as testing of these new sensors for possible performance improvements. This would include the currently available prepackaged systems for CubeSats discussed in Chapter I. The ultimate goal could be to develop a prepackaged ADS at NPS that could work with the proposed ACS developed in [31]. This would necessarily require implementation of the EKF developed in this thesis on a microprocessor like the MSP430 from TI.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX

A. ADDITIONAL SIMULATION RESULTS

1. Simulation 1

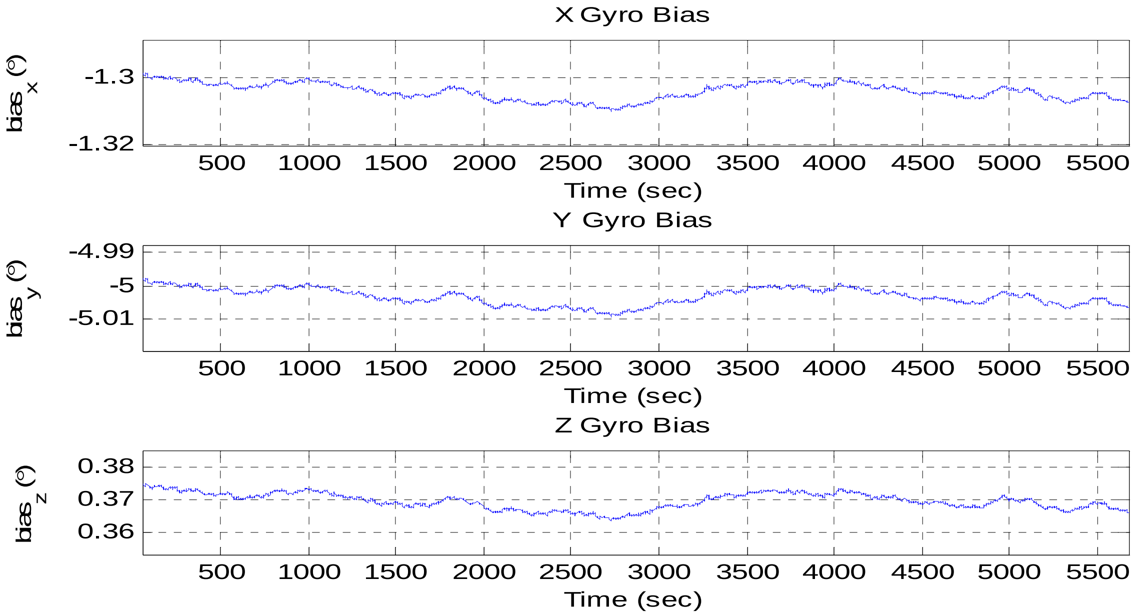


Figure 47. Simulation 1 Gyro Bias.

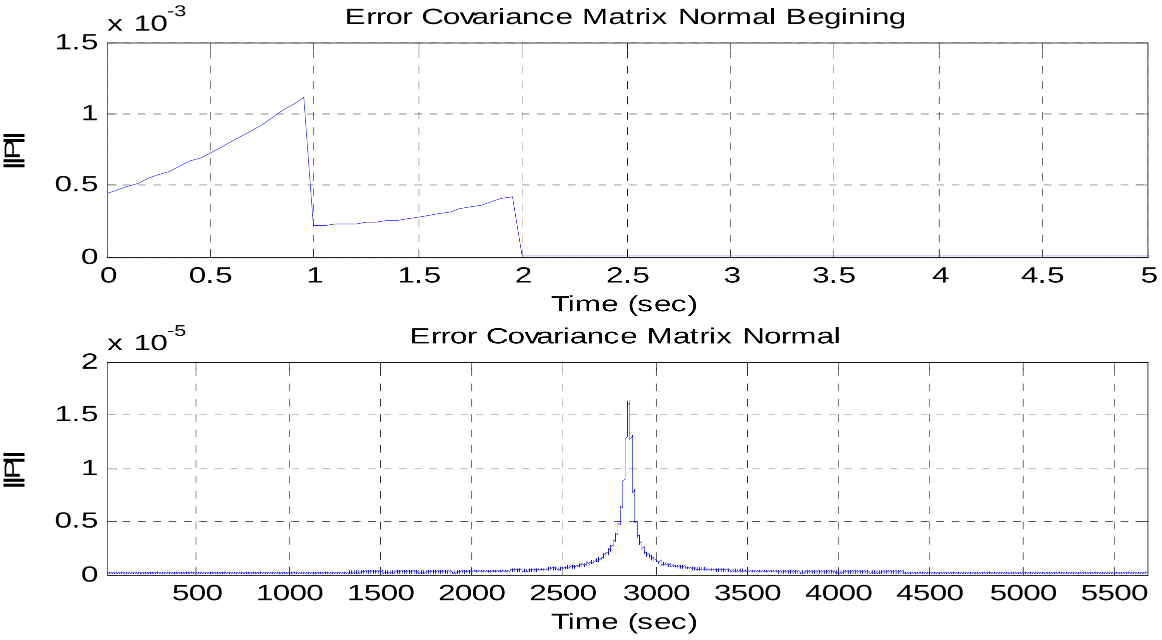


Figure 48. Simulation 1 Error Covariance Matrix Normal.

2. Simulation 2

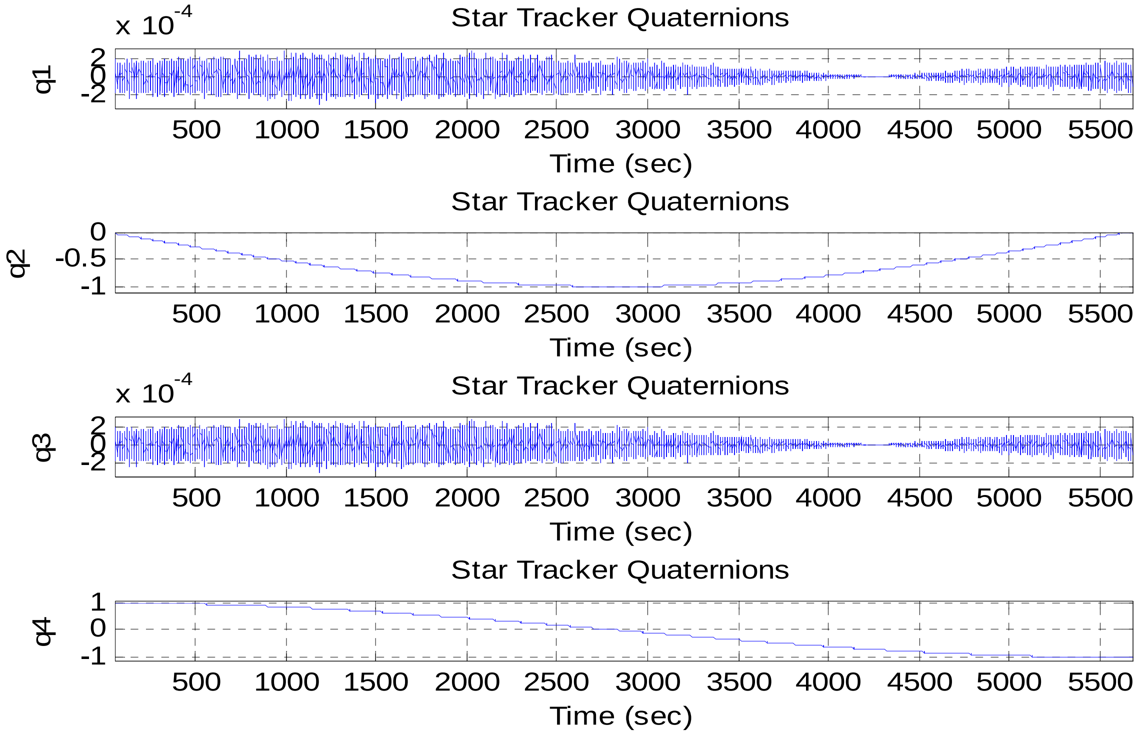


Figure 49. Simulation 2 Simulated Star Tracker Quaternion.

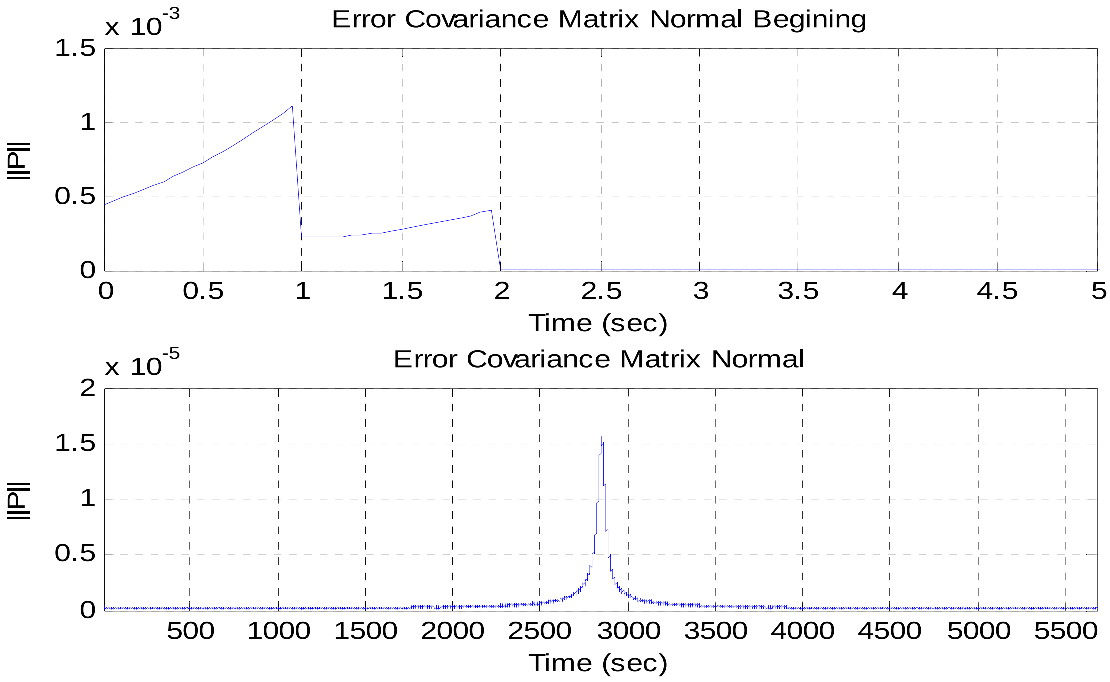


Figure 50. Simulation 2 Error Covariance Matrix Normal.

3. Simulation 3

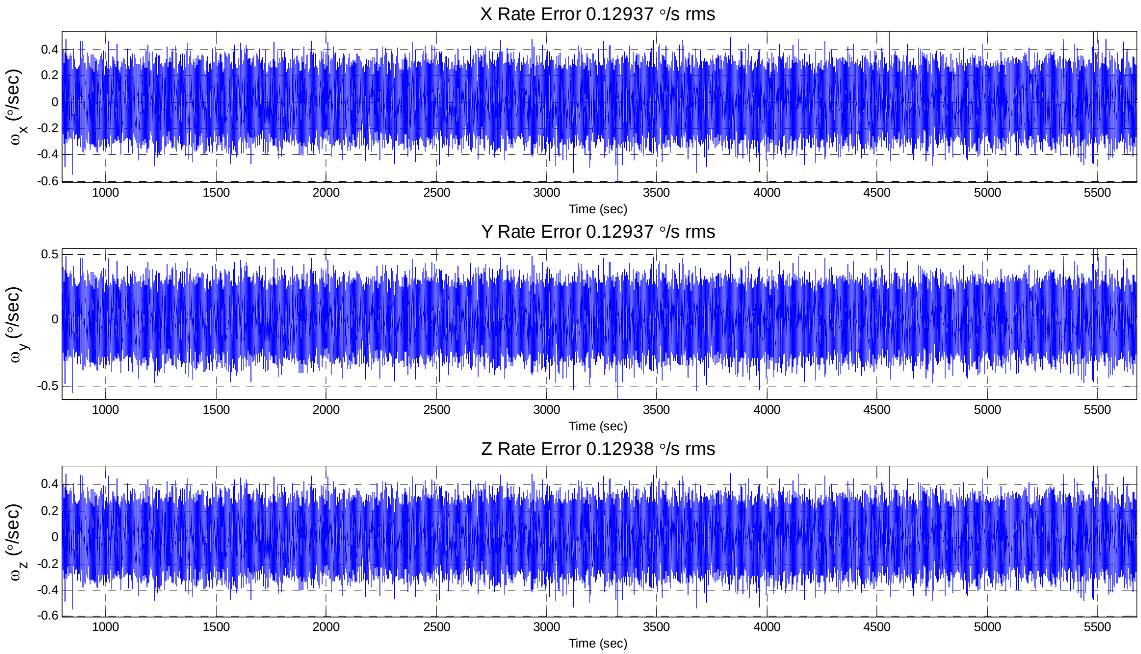


Figure 51. Simulation 3 Gyro Rate Error.

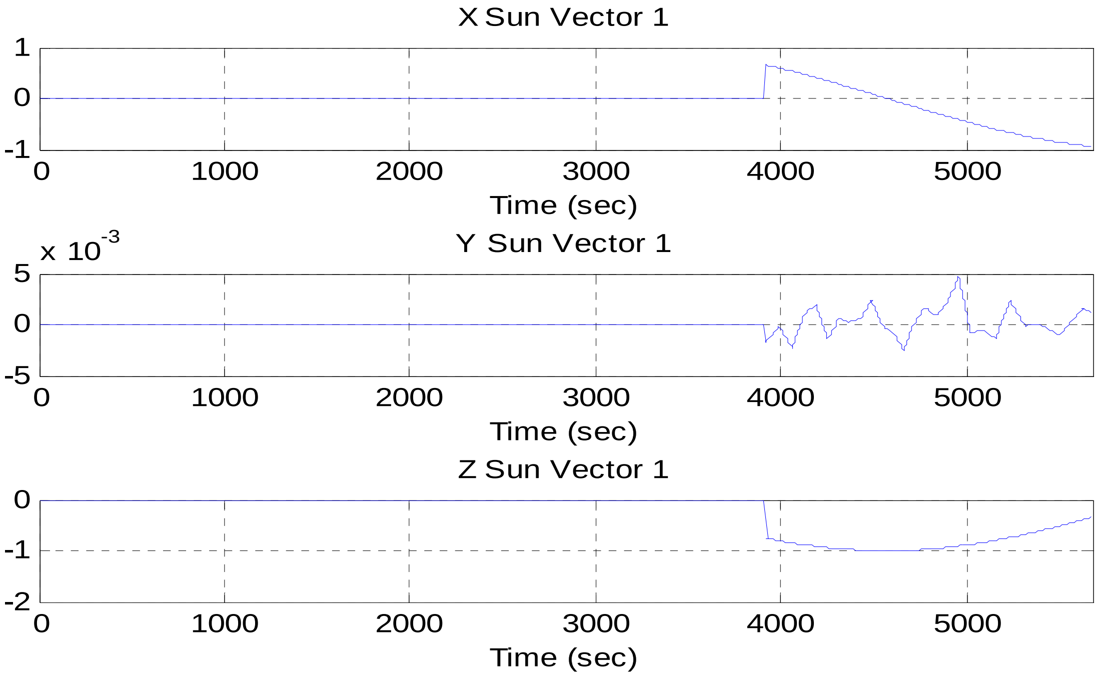


Figure 52. Simulation 3 Sun Sensor #1 Measurement.

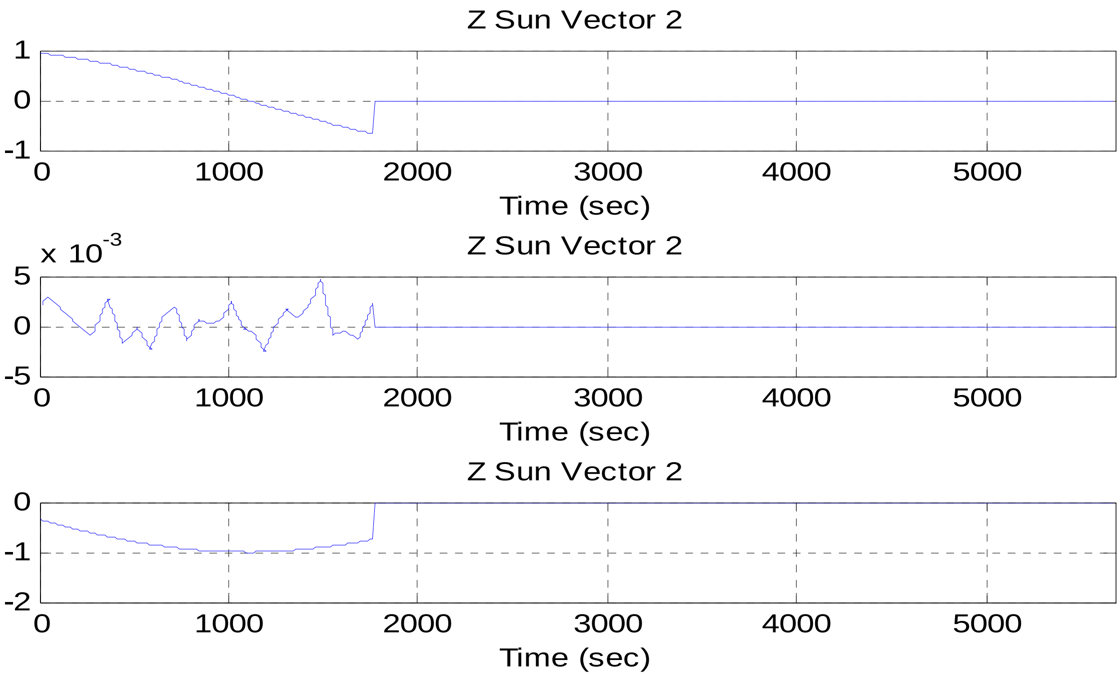


Figure 53. Simulation 3 Sun Sensor #2 Measurement.

4. Simulation 4

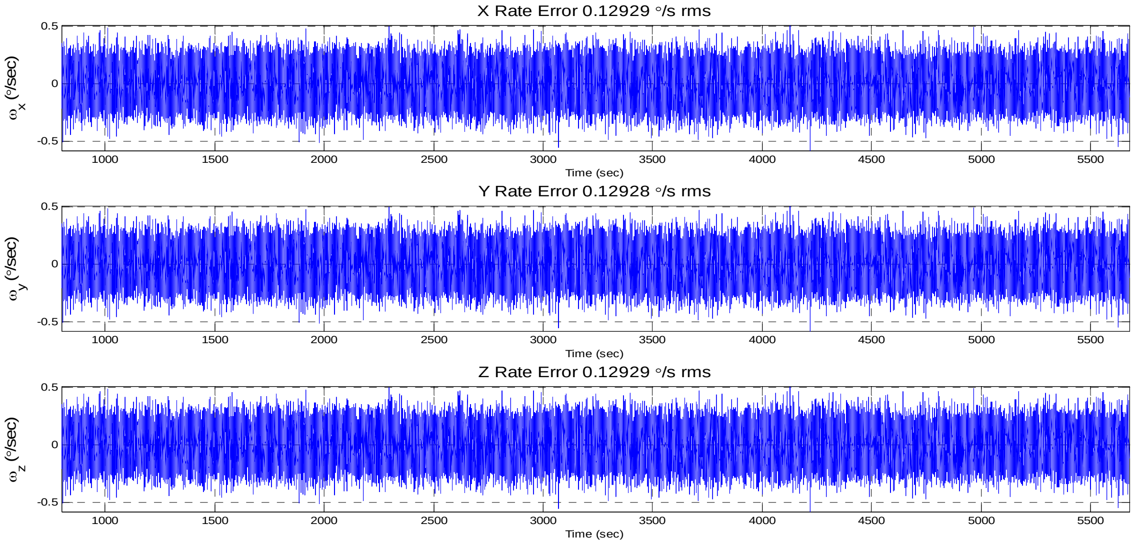


Figure 54. Simulation 4 Gyro Rate Error.

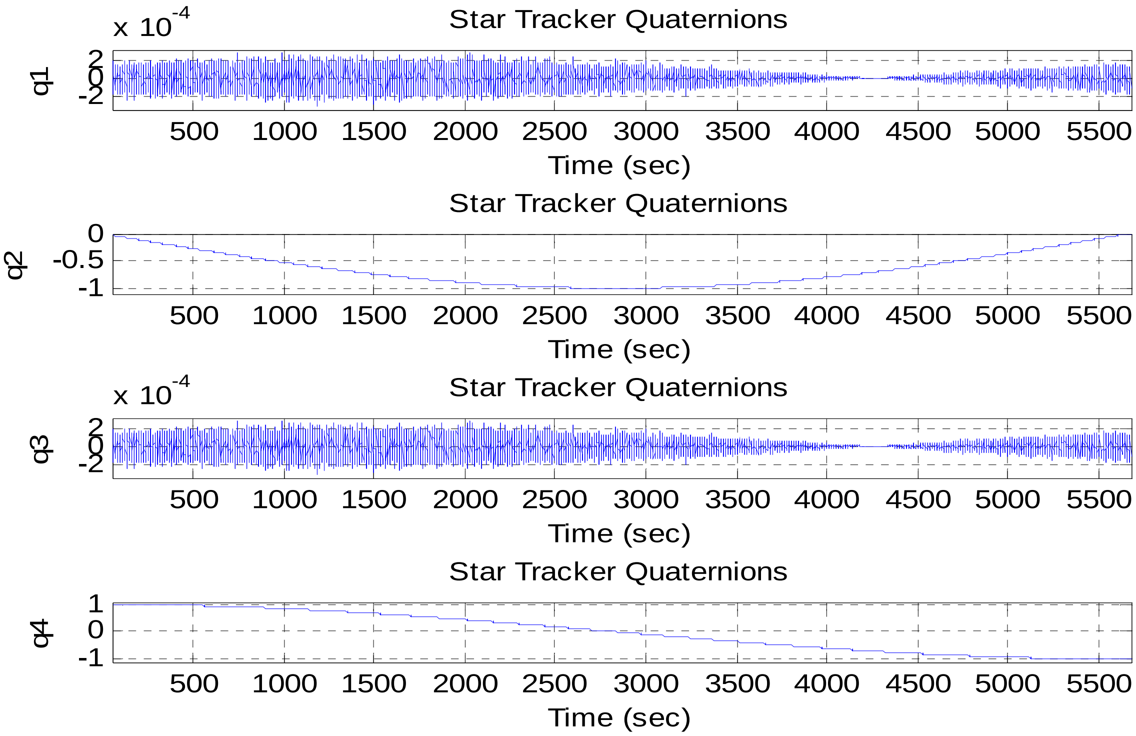


Figure 55. Simulation 4 Star tracker Quaternion Measurements.

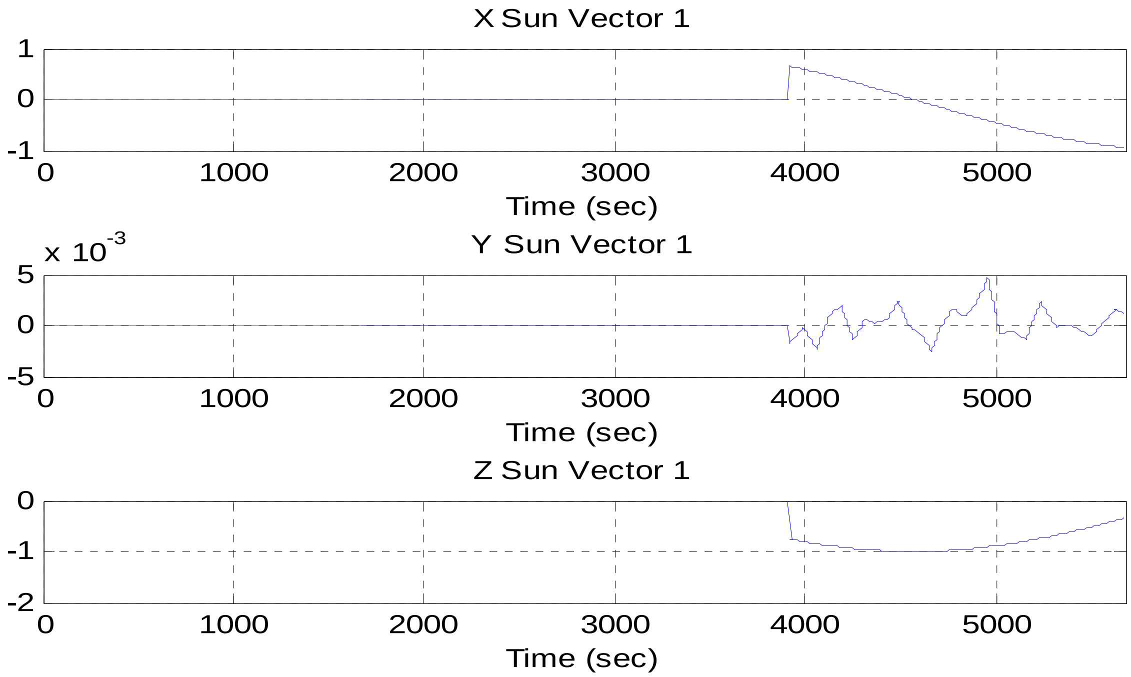


Figure 56. Simulation 4 Sun Sensor #1 Measurement.

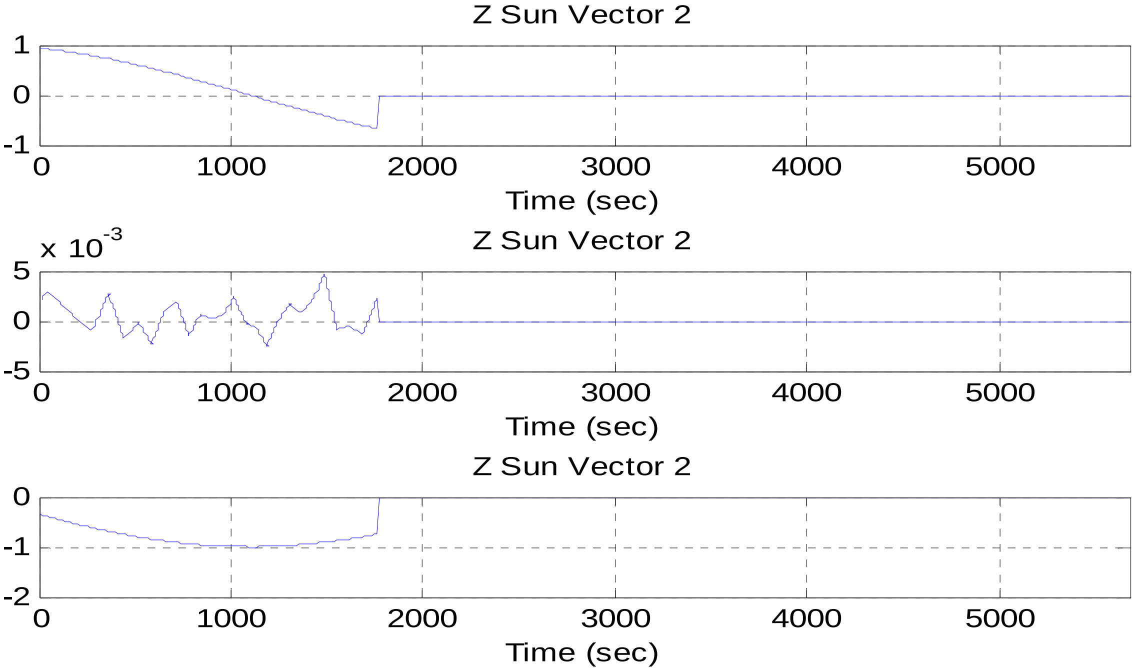


Figure 57. Simulation 4 Sun Sensor #2 Measurement.

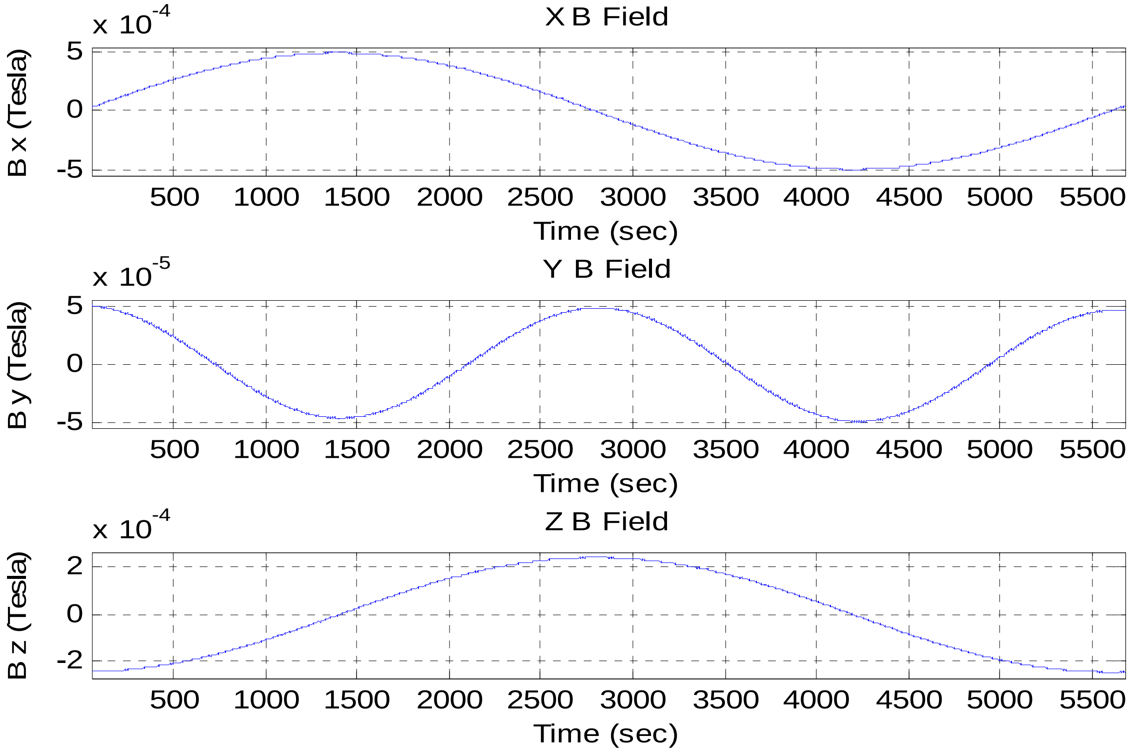
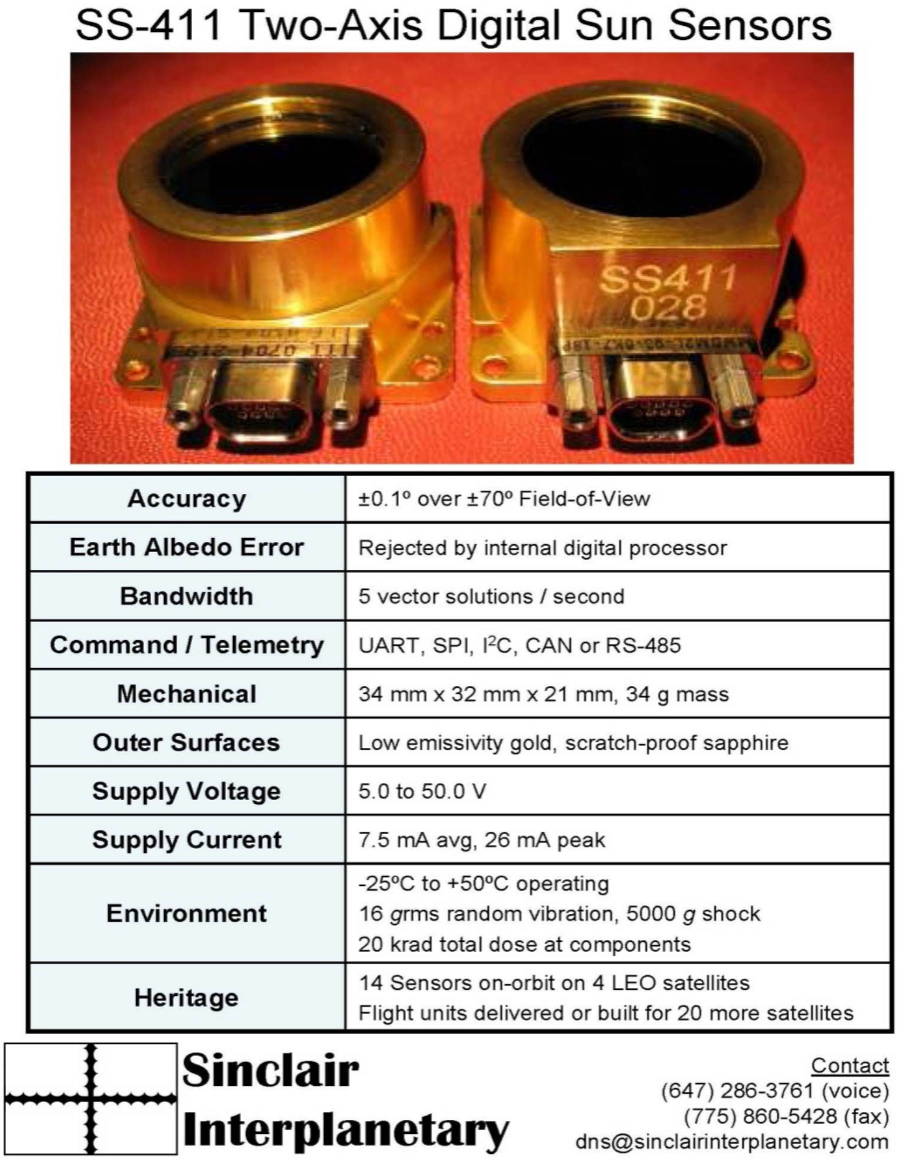


Figure 58. Simulation 4 Magnetometer Measurement.

B. SENSOR DATA SHEETS

This appendix contains all the data sheets for the selected sensors.

1. Sinclair Interplanetary SS-411



SS-411 Two-Axis Digital Sun Sensors

Accuracy	±0.1º over ±70º Field-of-View
Earth Albedo Error	Rejected by internal digital processor
Bandwidth	5 vector solutions / second
Command / Telemetry	UART, SPI, I^2C, CAN or RS-485
Mechanical	34 mm x 32 mm x 21 mm, 34 g mass
Outer Surfaces	Low emissivity gold, scratch-proof sapphire
Supply Voltage	5.0 to 50.0 V
Supply Current	7.5 mA avg, 26 mA peak
Environment	-25ºC to +50ºC operating 16 *g*rms random vibration, 5000 *g* shock 20 krad total dose at components
Heritage	14 Sensors on-orbit on 4 LEO satellites Flight units delivered or built for 20 more satellites

Sinclair Interplanetary

Contact
(647) 286-3761 (voice)
(775) 860-5428 (fax)
dns@sinclairinterplanetary.com

2. AeroAstro Mini Star Tracker

AEROASTRO

Miniature Star Tracker

AeroAstro's Miniature Star Tracker (MST) design is a small, low power (< 2 Watt) star tracker, with an accuracy of better than ±70 arc-seconds in all three axes (3σ). It achieves reasonable star tracking accuracy with low mass and power consumption at less than half the cost of other star trackers.

MST - Horizontal View

The MST is available in a low-cost commercial-off-the-shelf (COTS) version tolerant up to 30 krad and can be modified for higher radiation tolerance. The MST also provides a lost-in-space capability and is currently being enhanced to achieve fast angular rate sensing.

The star tracker features a user-definable star catalog and powerful hybrid processor. With a 1 Mpixel CMOS array, the star tracker is sensitive up to 4th magnitude stars.

MST - Vertical View

Images can be downloaded for ground processing and custom code can be incorporated. Built-in test includes the ability to upload images and verify star tracker performance.

Making Space For Everyone™

Specifications

Accuracy: Better than ±70 arc seconds, 3-axes (3σ)

CMOS Imager: ~ 1024 x 1280 pixel array each pixel ~ 7 µm square

Sensitivity: Up to 4th magnitude stars

Maximum Pitch/Yaw Rate: 10°/sec (goal)

Update Rate: ~ 1 Hz

Output: Quaternion, Centroids, and custom

Stars Tracked: Up to 9 simultaneously

Star Catalog: 600 and can employ user-defined catalogs

Image Rate: 0 to 24 fps

Power: < 2 Watts

Radiation Tolerance: Up to 30 krad(Si), more with shielding

Dimensions: 2”x 2”x 3”(5.4 cm x 5.4 cm x 7.6 cm)

Mass: 425 g (not including baffle)

Self Test: Images can be up and down loaded for verification

AEROASTRO
20145 Ashbrook Place
Ashburn, VA 20147 USA
info@aeroastro.com
www.aeroastro.com

t: 703.723.9800
f: 703.723.9850

Rev 07-02

3. NovAtel OEMV-1G

OEMV®-1 & OEMV®-1G

Features

L1 GPS+Integrated OmniSTAR and CDGPS on OEMV-1

L1 GPS+GLONASS on OEMV-1G

GL1DE technology

AdVance™ RTK Positioning

Application Programming Interface (API) option

Benefits

Sub-metre real-time accuracy

Additional GLONASS satellites offer increased position availability

Smooth, consistent positions for efficient pass-to-pass applications

20 cm real-time accuracy using NovAtel's proven RT-20™ technology

Reduces system hardware by using the receiver's processor and memory to run a user application

NovAtel®'s OEMV-1 products are high performance, single frequency GPS engines featuring additional signals and API support. The OEMV-1 offers integrated L1 GPS and OmniSTAR and CDGPS while the OEMV-1G offers L1 GPS+GLONASS, both in a compact size with low power consumption.

Designed for integration

The OEMV-1/1G receiver is designed for applications requiring accurate positioning performance in a small package. At just 46 millimetres by 71 millimetres and power consumption of 1.0 W, the OEMV-1/1G are extremely competitive precision L1 GPS engines. These flexible receivers provide the user with 20 centimetre real-time accuracy using NovAtel's AdVance RTK technology, smooth decimetre pass-to-pass performance using GL1DE™, or sub-metre accuracy from DGPS or L-band corrections.

OEMV-1 with integrated L-band

L-band functionality is integrated in the OEMV eliminating the need for additional hardware. Using the OEMV-1, users can access OmniSTAR VBS or CDGPS correction services for sub-metre accuracy, thus minimizing additional size, cost and complexity in the end user system. The OEMV-1 is the only L1 board available in the market with onboard L-band positioning.

OEMV-1G with GLONASS

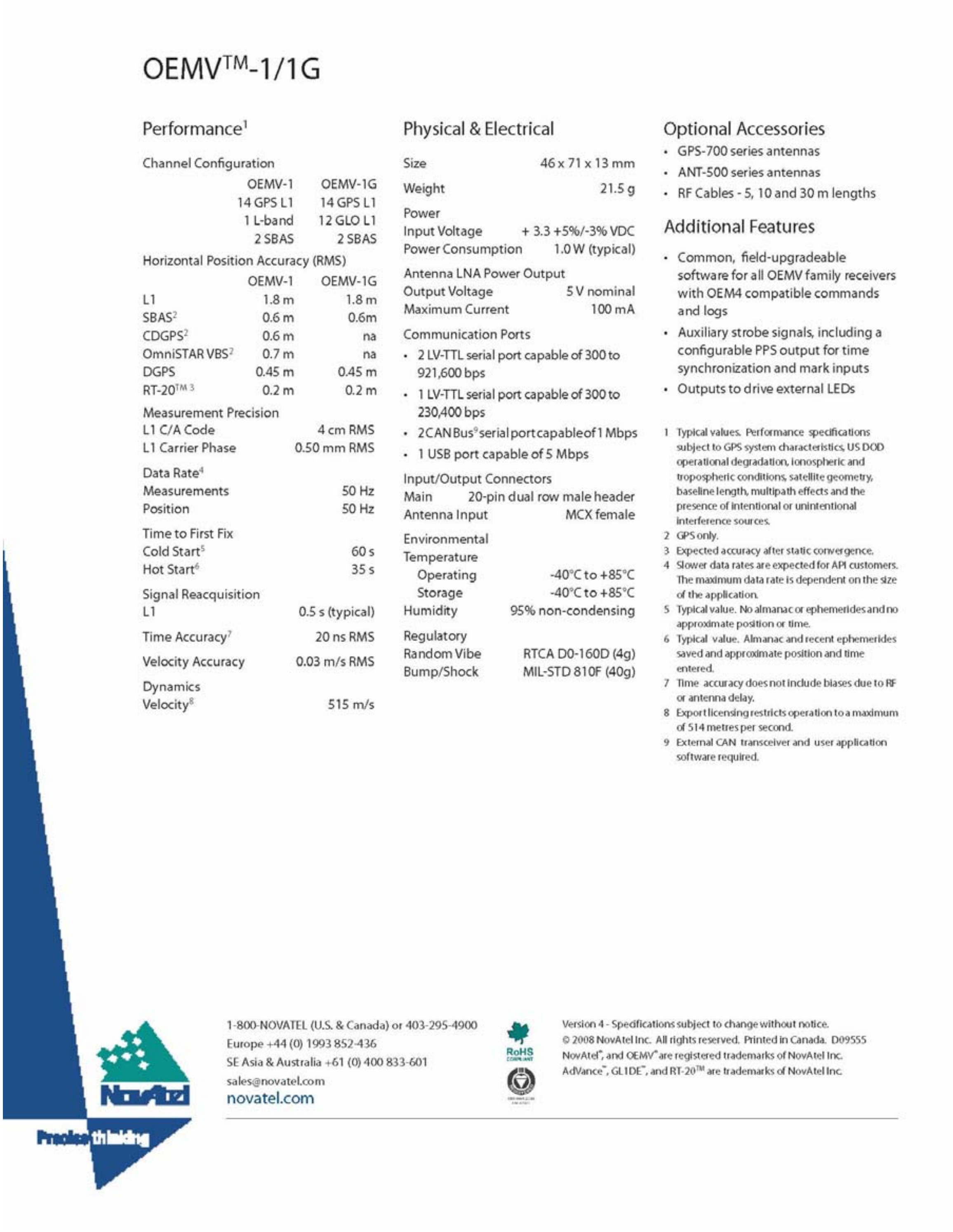
OEMV-1G receiver offers GPS+GLONASS positions and measurements that are used in combination with GPS data to provide more satellites for positioning in challenging environments. The OEMV-1G, OEMV-2 and OEMV-3 receivers and their enclosures are all configurable as either GPS only or GPS+GLONASS. The addition of GLONASS satellites to the solution enables users to work more often and increases availability of a position in obstructed sky conditions.

Customization with the API

The Application Programming Interface (API) functionality introduced on NovAtel's OEM4 receivers is also available on all OEMV receivers. With the API library and a standard C/C++ development environment, an application can be developed to run directly from the receiver platform, eliminating system hardware, reducing development time and resulting in faster time to market.

NovAtel

Precise thinking



OEMV™-1/1G

Performance[1]

Channel Configuration

	OEMV-1	OEMV-1G
	14 GPS L1	14 GPS L1
	1 L-band	12 GLO L1
	2 SBAS	2 SBAS

Horizontal Position Accuracy (RMS)

	OEMV-1	OEMV-1G
L1	1.8 m	1.8 m
SBAS[2]	0.6 m	0.6m
CDGPS[2]	0.6 m	na
OmniSTAR VBS[2]	0.7 m	na
DGPS	0.45 m	0.45 m
RT-20™[3]	0.2 m	0.2 m

Measurement Precision	
L1 C/A Code	4 cm RMS
L1 Carrier Phase	0.50 mm RMS
Data Rate[4]	
Measurements	50 Hz
Position	50 Hz
Time to First Fix	
Cold Start[5]	60 s
Hot Start[6]	35 s
Signal Reacquisition	
L1	0.5 s (typical)
Time Accuracy[7]	20 ns RMS
Velocity Accuracy	0.03 m/s RMS
Dynamics	
Velocity[8]	515 m/s

Physical & Electrical

Size	46 x 71 x 13 mm
Weight	21.5 g
Power	
Input Voltage	+ 3.3 +5%/-3% VDC
Power Consumption	1.0 W (typical)
Antenna LNA Power Output	
Output Voltage	5 V nominal
Maximum Current	100 mA

Communication Ports

- 2 LV-TTL serial port capable of 300 to 921,600 bps
- 1 LV-TTL serial port capable of 300 to 230,400 bps
- 2 CAN Bus[9] serial port capable of 1 Mbps
- 1 USB port capable of 5 Mbps

Input/Output Connectors	
Main	20-pin dual row male header
Antenna Input	MCX female
Environmental	
Temperature	
Operating	-40°C to +85°C
Storage	-40°C to +85°C
Humidity	95% non-condensing
Regulatory	
Random Vibe	RTCA D0-160D (4g)
Bump/Shock	MIL-STD 810F (40g)

Optional Accessories

- GPS-700 series antennas
- ANT-500 series antennas
- RF Cables - 5, 10 and 30 m lengths

Additional Features

- Common, field-upgradeable software for all OEMV family receivers with OEM4 compatible commands and logs
- Auxiliary strobe signals, including a configurable PPS output for time synchronization and mark inputs
- Outputs to drive external LEDs

[1] Typical values. Performance specifications subject to GPS system characteristics, US DOD operational degradation, ionospheric and tropospheric conditions, satellite geometry, baseline length, multipath effects and the presence of intentional or unintentional interference sources.

[2] GPS only.

[3] Expected accuracy after static convergence.

[4] Slower data rates are expected for API customers. The maximum data rate is dependent on the size of the application.

[5] Typical value. No almanac or ephemerides and no approximate position or time.

[6] Typical value. Almanac and recent ephemerides saved and approximate position and time entered.

[7] Time accuracy does not include biases due to RF or antenna delay.

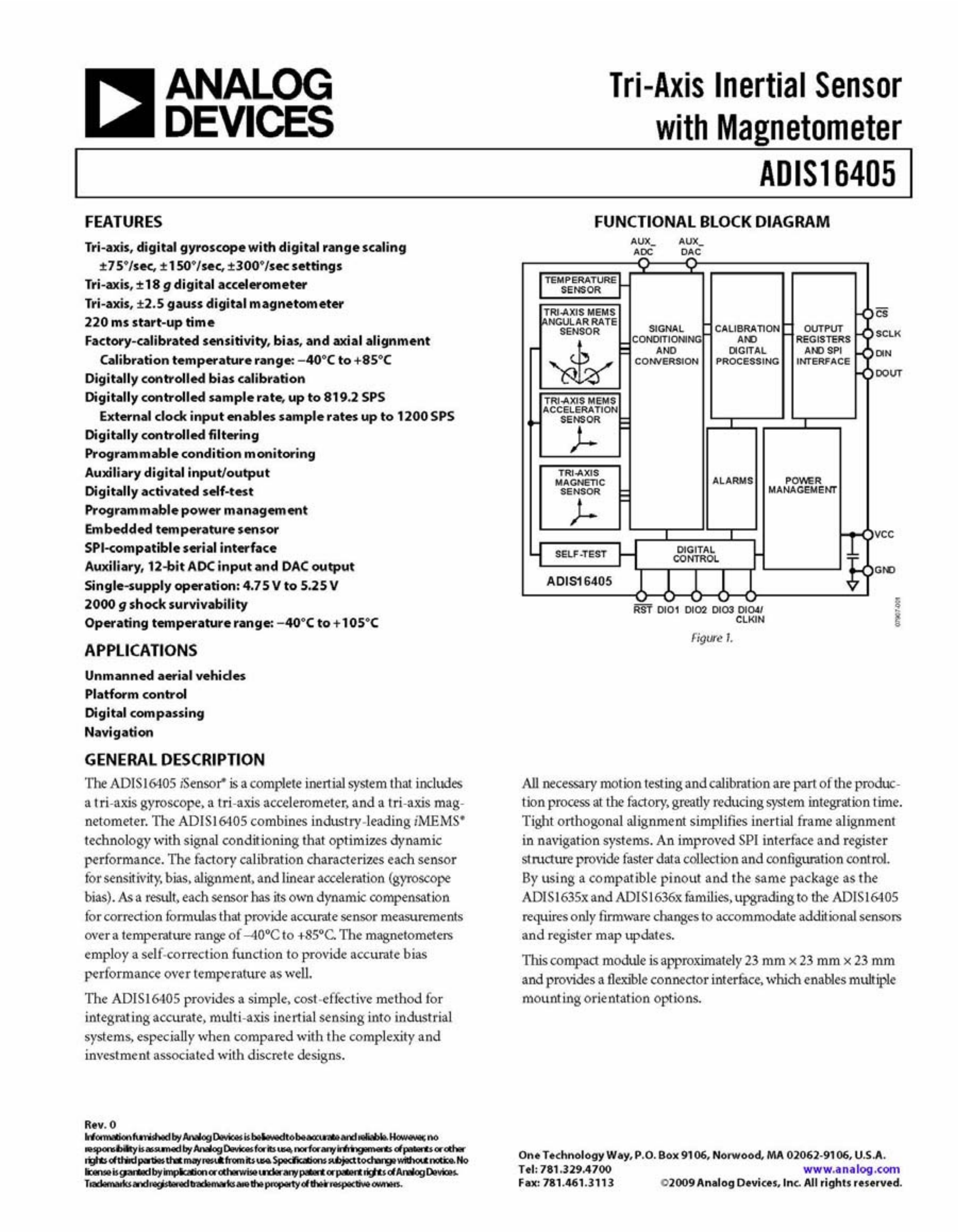
[8] Export licensing restricts operation to a maximum of 514 metres per second.

[9] External CAN transceiver and user application software required.

1-800-NOVATEL (U.S. & Canada) or 403-295-4900
Europe +44 (0) 1993 852-436
SE Asia & Australia +61 (0) 400 833-601
sales@novatel.com
novatel.com

Version 4 - Specifications subject to change without notice.

4. Analog Devices ADIS16405



Tri-Axis Inertial Sensor with Magnetometer

ADIS16405

FEATURES

Tri-axis, digital gyroscope with digital range scaling
±75°/sec, ±150°/sec, ±300°/sec settings
Tri-axis, ±18 *g* digital accelerometer
Tri-axis, ±2.5 gauss digital magnetometer
220 ms start-up time
Factory-calibrated sensitivity, bias, and axial alignment
Calibration temperature range: −40°C to +85°C
Digitally controlled bias calibration
Digitally controlled sample rate, up to 819.2 SPS
External clock input enables sample rates up to 1200 SPS
Digitally controlled filtering
Programmable condition monitoring
Auxiliary digital input/output
Digitally activated self-test
Programmable power management
Embedded temperature sensor
SPI-compatible serial interface
Auxiliary, 12-bit ADC input and DAC output
Single-supply operation: 4.75 V to 5.25 V
2000 *g* shock survivability
Operating temperature range: −40°C to +105°C

APPLICATIONS

Unmanned aerial vehicles
Platform control
Digital compassing
Navigation

FUNCTIONAL BLOCK DIAGRAM

Figure 1.

GENERAL DESCRIPTION

The ADIS16405 *i*Sensor® is a complete inertial system that includes a tri-axis gyroscope, a tri-axis accelerometer, and a tri-axis magnetometer. The ADIS16405 combines industry-leading *i*MEMS® technology with signal conditioning that optimizes dynamic performance. The factory calibration characterizes each sensor for sensitivity, bias, alignment, and linear acceleration (gyroscope bias). As a result, each sensor has its own dynamic compensation for correction formulas that provide accurate sensor measurements over a temperature range of −40°C to +85°C. The magnetometers employ a self-correction function to provide accurate bias performance over temperature as well.

The ADIS16405 provides a simple, cost-effective method for integrating accurate, multi-axis inertial sensing into industrial systems, especially when compared with the complexity and investment associated with discrete designs.

All necessary motion testing and calibration are part of the production process at the factory, greatly reducing system integration time. Tight orthogonal alignment simplifies inertial frame alignment in navigation systems. An improved SPI interface and register structure provide faster data collection and configuration control. By using a compatible pinout and the same package as the ADIS1635x and ADIS1636x families, upgrading to the ADIS16405 requires only firmware changes to accommodate additional sensors and register map updates.

This compact module is approximately 23 mm × 23 mm × 23 mm and provides a flexible connector interface, which enables multiple mounting orientation options.

Rev. 0

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 **www.analog.com**
Fax: 781.461.3113

TABLE OF CONTENTS

REVISION HISTORY

3/09—Revision 0: Initial Version

SPECIFICATIONS

$T_A = -40°C$ to $+85°C$, VCC = 5.0 V, angular rate = 0°/sec, dynamic range = ±300°/sec, ±1 *g*, unless otherwise noted.

Table 1.

Parameter	Test Conditions	Min	Typ	Max	Unit
GYROSCOPES					
Dynamic Range		±300	±350		°/sec
Initial Sensitivity	Dynamic range = ±300°/sec	0.0495	0.05	0.0505	°/sec/LSB
	Dynamic range = ±150°/sec		0.025		°/sec/LSB
	Dynamic range = ±75°/sec		0.0125		°/sec/LSB
Sensitivity Temperature Coefficient	$-40°C \le T_A \le +85°C$		±40		ppm/°C
Misalignment	Axis-to-axis, Δ = 90° ideal		±0.05		Degrees
	Axis-to-frame (package)		±0.5		Degrees
Nonlinearity	Best fit straight line		0.1		% of FS
Initial Bias Error	1 σ		±3		°/sec
In-Run Bias Stability	1 σ, SMPL_PRD = 0x01		0.007		°/sec
Angular Random Walk	1 σ, SMPL_PRD = 0x01		2.0		°/√hr
Bias Temperature Coefficient	$-40°C \le T_A \le +85°C$		±0.01		°/sec/°C
Linear Acceleration Effect on Bias	Any axis, 1 σ (MSC_CTRL, Bit 7 = 1)		0.05		°/sec/*g*
Bias Voltage Sensitivity	VCC = 4.75 V to 5.25 V		0.32		°/sec/V
Output Noise	±300°/sec range, no filtering		0.9		°/sec rms
Rate Noise Density	f = 25 Hz, ±300°/sec, no filtering		0.05		°/sec/√Hz rms
3 dB Bandwidth			330		Hz
ACCELEROMETERS					
Dynamic Range		±18			*g*
Initial Sensitivity		3.285	3.33	3.38	m*g*/LSB
Sensitivity Temperature Coefficient	$-40°C \le T_A \le +85°C$		±50		ppm/°C
Misalignment	Axis-to-axis, Δ = 90° ideal		0.2		Degrees
	Axis-to-frame (package)		±0.5		Degrees
Nonlinearity	Best fit straight line, ±17 *g*		0.1		% of FS
Initial Bias Error	1 σ		±50		m*g*
In-Run Bias Stability	1 σ		0.2		m*g*
Velocity Random Walk	1 σ		0.2		m/sec/√hr
Bias Temperature Coefficient	$-40°C \le T_A \le +85°C$		±0.3		m*g*/°C
Bias Voltage Sensitivity	VCC = 4.75 V to 5.25 V		2.5		m*g*/V
Output Noise	No filtering		9		m*g* rms
Noise Density	No filtering		0.5		m*g*/√Hz rms
3 dB Bandwidth			330		Hz
MAGNETOMETER					
Dynamic Range		±2.5	±3.5		gauss
Initial Sensitivity	25°C	0.49	0.5	0.51	mgauss/LSB
Sensitivity Temperature Coefficient	25°C, 1 σ		600		ppm/°C
Axis Nonorthogonality	25°C, axis-to-axis		0.25		Degrees
Axis Misalignment	25°C, axis-to-base plate and guide pins		0.5		Degrees
Nonlinearity	Best fit straight line		0.5		% of FS
Initial Bias Error	25°C, 0 gauss stimulus		±4		mgauss
Bias Temperature Coefficient			0.5		mgauss/°C
Output Noise	25°C, no filtering		1.25		mgauss rms
Noise Density	25°C, no filtering, rms		0.066		mgauss/√Hz
3 dB Bandwidth			1540		Hz
TEMPERATURE SENSOR					
Scale Factor	25°C, output = 0x0000		0.14		°C/LSB

Parameter	Test Conditions	Min	Typ	Max	Unit
ADC INPUT					
Resolution			12		Bits
Integral Nonlinearity			±2		LSB
Differential Nonlinearity			±1		LSB
Offset Error			±4		LSB
Gain Error			±2		LSB
Input Range		0		3.3	V
Input Capacitance	During acquisition		20		pF
DAC OUTPUT					
Resolution			12		Bits
Relative Accuracy	Code 101 to Code 4095, 5 kΩ/100 pF to GND		±4		LSB
Differential Nonlinearity			±1		LSB
Offset Error			±5		mV
Gain Error			±0.5		%
Output Range		0		3.3	V
Output Impedance			2		Ω
Output Settling Time	5 kΩ/100 pF to GND		10		μs
LOGIC INPUTS[1]					
Input High Voltage, V_{INH}		2.0			V
Input Low Voltage, V_{INL}				0.8	V
	$\overline{CS}$ signal to wake up from sleep mode			0.55	V
$\overline{CS}$ Wake-Up Pulse Width		20			μs
Logic 1 Input Current, I_{INH}	V_{IH} = 3.3 V		±0.2	±10	μA
Logic 0 Input Current, I_{INL}	V_{IL} = 0 V				
All Pins Except $\overline{RST}$			−40	−60	μA
$\overline{RST}$ Pin			−1		mA
Input Capacitance, C_{IN}			10		pF
DIGITAL OUTPUTS[1]					
Output High Voltage, V_{OH}	I_{SOURCE} = 1.6 mA	2.4			V
Output Low Voltage, V_{OL}	I_{SINK} = 1.6 mA			0.4	V
FLASH MEMORY	Endurance[2]	10,000			Cycles
Data Retention[3]	T_J = 85°C	10			Years
FUNCTIONAL TIMES[4]	Time until data is available				
Power-On Start-Up Time	Normal mode, SMPL_PRD ≤ 0x09		220		ms
	Low power mode, SMPL_PRD ≥ 0x0A		290		ms
Reset Recovery Time	Normal mode, SMPL_PRD ≤ 0x09		100		ms
	Low power mode, SMPL_PRD ≥ 0x0A		170		ms
Sleep Mode Recovery Time	Normal mode, SMPL_PRD ≤ 0x09		4		ms
	Low power mode, SMPL_PRD ≥ 0x0A		15		
Flash Memory Test Time	Normal mode, SMPL_PRD ≤ 0x09		17		ms
	Low power mode, SMPL_PRD ≥ 0x0A		90		ms
Automatic Self-Test Time	SMPL_PRD = 0x01		12		ms
CONVERSION RATE	SMPL_PRD = 0x01 to 0xFF	0.413		819.2	SPS
Clock Accuracy				±3	%
Sync Input Clock				1.2	kHz
POWER SUPPLY					
Operating Voltage Range, VCC		4.75	5.0	5.25	V
Power Supply Current	Low power mode at 25°C		45		mA
	Normal mode at 25°C		70		mA
	Sleep mode at 25°C		600		μA

---

[1] The digital I/O signals are driven by an internal 3.3 V supply, and the inputs are 5 V tolerant.

[2] Endurance is qualified as per JEDEC Standard 22, Method A117, and measured at −40°C, +25°C, +85°C, and +125°C.

[3] The data retention lifetime equivalent is at a junction temperature (T_J) of 85°C as per JEDEC Standard 22, Method A117. Data retention lifetime decreases with junction temperature.

[4] These times do not include thermal settling and internal filter response times (330 Hz bandwidth), which may affect overall accuracy.

TIMING SPECIFICATIONS

T_A = 25°C, VCC = 5 V, unless otherwise noted.

Table 2.

Parameter	Description	Normal Mode (SMPL_PRD ≤ 0x09)			Low Power Mode (SMPL_PRD ≥ 0x0A)			Burst Mode			Unit
		Min[1]	Typ	Max	Min[1]	Typ	Max	Min[1]	Typ	Max	
f_{SCLK}		0.01		2.0	0.01		0.3	0.01		1.0	MHz
t_{STALL}	Stall period between data	9			75			$1/f_{SCLK}$			μs
$t_{READRATE}$	Read rate	40			150						μs
t_{CS}	Chip select to clock edge	48.8			48.8			48.8			ns
t_{DAV}	DOUT valid after SCLK edge			100			100			100	ns
t_{DSU}	DIN setup time before SCLK rising edge	24.4			24.4			24.4			ns
t_{DHD}	DIN hold time after SCLK rising edge	48.8			48.8			48.8			ns
t_{SCLKR}, t_{SCLKF}	SCLK rise/fall times		5	12.5		5	12.5		5	12.5	ns
t_{DF}, t_{DR}	DOUT rise/fall times		5	12.5		5	12.5		5	12.5	ns
t_{SFS}	$\overline{CS}$ high after SCLK edge	5			5			5			ns
t_1	Input sync pulse width		5								μs
t_2	Input sync to data ready output		600								μs
t_3	Input sync period	833									μs

---

[1] Guaranteed by design and characterization, but not tested in production.

TIMING DIAGRAMS

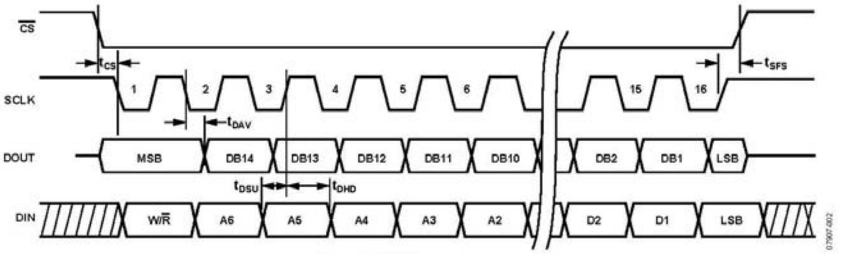


Figure 2. SPI Timing and Sequence

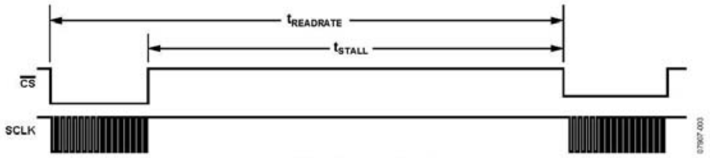


Figure 3. Stall Time and Data Rate

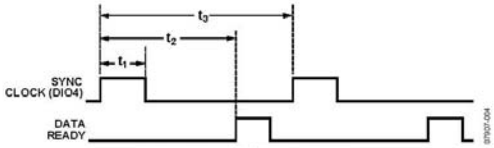


Figure 4. Input Clock Timing Diagram

ABSOLUTE MAXIMUM RATINGS

Table 3.

Parameter	Rating
Acceleration	
Any Axis, Unpowered	2000 *g*
Any Axis, Powered	2000 *g*
VCC to GND	−0.3 V to +6.0 V
Digital Input Voltage to GND	−0.3 V to +5.3 V
Digital Output Voltage to GND	−0.3 V to VCC + 0.3 V
Analog Input to GND	−0.3 V to +3.6 V
Operating Temperature Range	−40°C to +105°C
Storage Temperature Range	−65°C to +125°C[1, 2]

[1] Extended exposure to temperatures outside the specified temperature range of −40°C to +105°C can adversely affect the accuracy of the factory calibration. For best accuracy, store the parts within the specified operating range of −40°C to +105°C.

[2] Although the device is capable of withstanding short-term exposure to 150°C, long-term exposure threatens internal mechanical integrity.

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

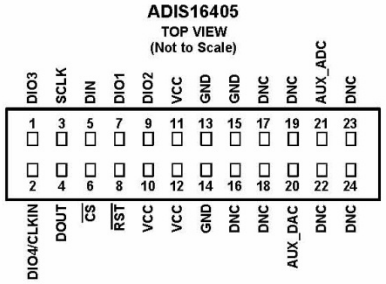
Table 4. Package Characteristics

Package Type	θ_{JA}	θ_{JC}	Device Weight
24-Lead Module	39.8°C/W	14.2°C/W	16 grams

ESD CAUTION

ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

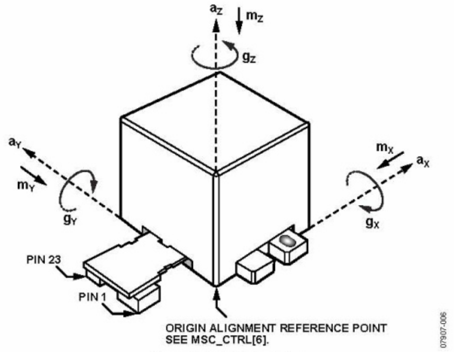
PIN CONFIGURATION AND FUNCTION DESCRIPTIONS



NOTES
1. THIS VIEW REPRESENTS THE TOP VIEW OF THE MATING CONNECTOR.
2. WHEN CONNECTED TO THE ADIS16405, THE PINS WILL NOT BE VISIBLE.
3. MATING CONNECTOR: SAMTEC CLM-112-02 OR EQUIVALENT.
4. DNC = DO NOT CONNECT.

07907-005

Figure 5. Pin Configuration



ORIGIN ALIGNMENT REFERENCE POINT
SEE MSC_CTRL[6].

Figure 6. Axial Orientation

Table 5. Pin Function Descriptions

Pin No.	Mnemonic	Type[1]	Description
1	DIO3	I/O	Configurable Digital Input/Output.
2	DIO4/CLKIN	I/O	Configurable Digital Input/Output or Sync Clock Input.
16, 17, 18, 19, 22, 23, 24	DNC	N/A	Do Not Connect.
3	SCLK	I	SPI Serial Clock.
4	DOUT	O	SPI Data Output. Clocks output on SCLK falling edge.
5	DIN	I	SPI Data Input. Clocks input on SCLK rising edge.
6	$\overline{CS}$	I	SPI Chip Select.
7	DIO1	I/O	Configurable Digital Input/Output.
8	$\overline{RST}$	I	Reset.
9	DIO2	I/O	Configurable Digital Input/Output.
10, 11, 12	VCC	S	Power Supply.
13, 14, 15	GND	S	Power Ground.
20	AUX_DAC	O	Auxiliary, 12-Bit DAC Output.
21	AUX_ADC	I	Auxiliary, 12-Bit ADC Input.

[1] S is supply, O is output, I is input, N/A is not applicable.

ADIS16405

TYPICAL PERFORMANCE CHARACTERISTICS

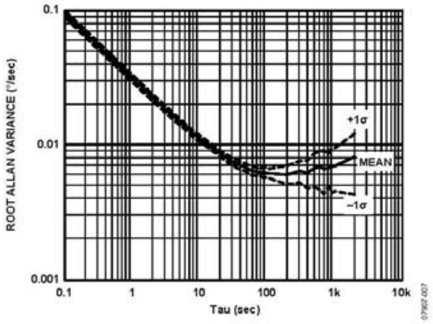


Figure 7. Gyroscope Root Allan Variance

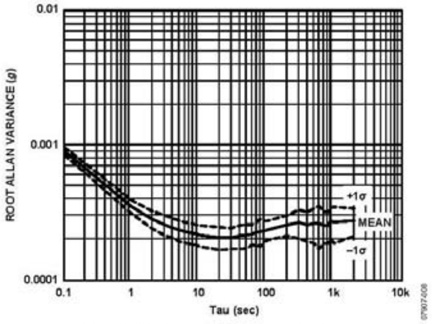


Figure 8. Accelerometer Root Allan Variance

Rev. 0 | Page 8 of 16

THEORY OF OPERATION

BASIC OPERATION

The ADIS16405 is an autonomous sensor system that starts up after a valid power supply voltage is applied and then begins producing inertial measurement data at the factory-default sample rate of 819.2 SPS. After each sample cycle, the sensor data loads into the output registers and DIO1 pulses, providing a new data ready control signal for driving system-level interrupt service routines. In a typical system, a master processor accesses the output data registers through the SPI interface, using the hook-up shown in Figure 9. Table 6 provides a generic functional description for each pin on the master processor. Table 7 describes the typical master processor settings normally found in a configuration register and used for communicating with the ADIS16405.

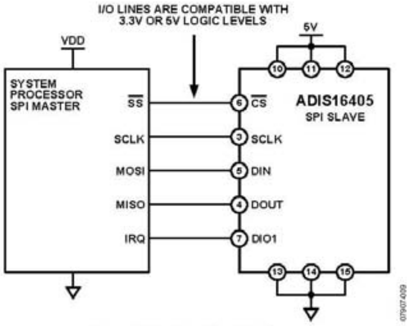


Figure 9. Electrical Hook-Up Diagram

Table 6. Generic Master Processor Pin Names and Functions

Pin Name	Function
$\overline{\text{SS}}$	Slave select
IRQ	Interrupt request
MOSI	Master output, slave input
MISO	Master input, slave output
SCLK	Serial clock

Table 7. Generic Master Processor SPI Settings

Processor Setting	Description
Master	The ADIS16405 operates as a slave.
SCLK Rate ≤ 2 MHz[1]	Normal mode, SMPL_PRD[7:0] ≤ 0x08.
CPOL = 1	Clock polarity.
CPHA = 1	Clock phase.
MSB-First	Bit sequence.
16-Bit	Shift register/data length.

[1] For burst mode, SCLK rate ≤ 1 MHz. For low power mode, SCLK rate ≤ 300 kHz.

The user registers provide addressing for all input/output operations on the SPI interface. Each 16-bit register has two 7-bit addresses: one for its upper byte and one for its lower byte. Table 8 lists the lower byte address for each register, and Figure 10 shows the generic bit assignments.

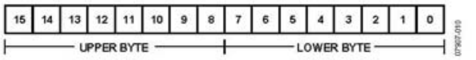


Figure 10. Output Register Bit Assignments

READING SENSOR DATA

Although the ADIS16405 produces data independently, it operates as an SPI slave device that communicates with system (master) processors using the 16-bit segments displayed in Figure 11. Individual register reads require two such 16-bit sequences. The first 16-bit sequence provides the read command bit (R/W = 0) and the target register address (A6 to A0). The second sequence transmits the register contents (D15 to D0) on the DOUT line. For example, if DIN = 0x0A00, then the content of XACCL_OUT shifts out on the DOUT line during the next 16-bit sequence.

The SPI operates in full duplex mode, which means that the master processor can read the output data from DOUT while using the same SCLK pulses to transmit the next target address on DIN.

DEVICE CONFIGURATION

The user register memory map (Table 8) identifies configuration registers with either a W or R/W. Configuration commands also use the bit sequence displayed in Figure 12. If the MSB is equal to 1, the last eight bits (DC7 to DC0) in the DIN sequence load into the memory address associated with the address bits (A5 to A0). For example, if DIN = 0xA11F, then 0x1F loads into Address Location 0x21 (XACCL_OFF, upper byte) at the conclusion of the data frame.

Most of the registers have a backup location in nonvolatile flash memory. The master processor must manage the backup function. Set GLOB_CMD[3] = 1 (DIN = 0xBE04) to execute a manual flash update (backup) operation, which copies the user registers into their respective flash memory locations. This operation takes 50 ms and requires the power supply voltage to be within the specified limit to complete properly. The FLASH_CNT register provides a running count of these events for managing the long-term reliability of the flash memory.

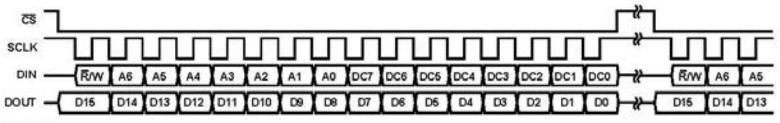
BURST MODE DATA COLLECTION

Burst mode data collection offers a more efficient method for collecting data from the ADIS16405. In sequential data cycles (each separated by one SCLK period), all output registers clock out on DOUT. This sequence starts when the DIN sequence is 0011 1110 0000 0000 (0x3E00). Next, the contents of each output register are output from DOUT, starting with SUPPLY_OUT and ending with AUX_ADC (see Figure 12). The addressing sequence shown in Table 8 determines the order of the outputs in burst mode.

Table 8. User Register Memory Map

Name	R/W	Flash Backup	Address[1]	Default	Function	Bit Assignments
FLASH_CNT	R	Yes	0x00	N/A	Flash memory write count	N/A
SUPPLY_OUT	R	No	0x02	N/A	Power supply measurement	Table 9
XGYRO_OUT	R	No	0x04	N/A	X-axis gyroscope output	Table 9
YGYRO_OUT	R	No	0x06	N/A	Y-axis gyroscope output	Table 9
ZGYRO_OUT	R	No	0x08	N/A	Z-axis gyroscope output	Table 9
XACCL_OUT	R	No	0x0A	N/A	X-axis accelerometer output	Table 9
YACCL_OUT	R	No	0x0C	N/A	Y-axis accelerometer output	Table 9
ZACCL_OUT	R	No	0x0E	N/A	Z-axis accelerometer output	Table 9
XMAGN_OUT	R	No	0x10	N/A	X-axis magnetometer measurement	Table 9
YMAGN_OUT	R	No	0x12	N/A	Y-axis magnetometer measurement	Table 9
ZMAGN_OUT	R	No	0x14	N/A	Z-axis magnetometer measurement	Table 9
TEMP_OUT	R	No	0x16	N/A	Temperature output	Table 9
AUX_ADC	R	No	0x18	N/A	Auxiliary ADC measurement	Table 9
XGYRO_OFF	R/W	Yes	0x1A	0x0000	X-axis gyroscope bias offset factor	Table 10
YGYRO_OFF	R/W	Yes	0x1C	0x0000	Y-axis gyroscope bias offset factor	Table 10
ZGYRO_OFF	R/W	Yes	0x1E	0x0000	Z-axis gyroscope bias offset factor	Table 10
XACCL_OFF	R/W	Yes	0x20	0x0000	X-axis acceleration bias offset factor	Table 11
YACCL_OFF	R/W	Yes	0x22	0x0000	Y-axis acceleration bias offset factor	Table 11
ZACCL_OFF	R/W	Yes	0x24	0x0000	Z-axis acceleration bias offset factor	Table 11
XMAGN_HIF	R/W	Yes	0x26	0x0000	X-axis magnetometer, hard-iron factor	Table 12
YMAGN_HIF	R/W	Yes	0x28	0x0000	Y-axis magnetometer, hard-iron factor	Table 12
ZMAGN_HIF	R/W	Yes	0x2A	0x0000	Z-axis magnetometer, hard-iron factor	Table 12
XMAGN_SIF	R/W	Yes	0x2C	0x0000	X-axis magnetometer, soft-iron factor	Table 13
YMAGN_SIF	R/W	Yes	0x2E	0x0000	Y-axis magnetometer, soft-iron factor	Table 13
ZMAGN_SIF	R/W	Yes	0x30	0x0000	Z-axis magnetometer, soft-iron factor	Table 13
GPIO_CTRL	R/W	No	0x32	0x0000	Auxiliary digital input/output control	Table 18
MSC_CTRL	R/W	Yes	0x34	0x0006	Miscellaneous control	Table 19
SMPL_PRD	R/W	Yes	0x36	0x0001	Internal sample period (rate) control	Table 15
SENS_AVG	R/W	Yes	0x38	0x0402	Dynamic range and digital filter control	Table 17
SLP_CNT	W	No	0x3A	0x0000	Sleep mode control	Table 16
DIAG_STAT	R	No	0x3C	0x0000	System status	Table 23
GLOB_CMD	W	N/A	0x3E	0x0000	System command	Table 14
ALM_MAG1	R/W	Yes	0x40	0x0000	Alarm 1 amplitude threshold	Table 25
ALM_MAG2	R/W	Yes	0x42	0x0000	Alarm 2 amplitude threshold	Table 25
ALM_SMPL1	R/W	Yes	0x44	0x0000	Alarm 1 sample size	Table 26
ALM_SMPL2	R/W	Yes	0x46	0x0000	Alarm 2 sample size	Table 26
ALM_CTRL	R/W	Yes	0x48	0x0000	Alarm control	Table 24
AUX_DAC	R/W	No	0x4A	0x0000	Auxiliary DAC data	Table 20

[1] Each register contains two bytes. The address of the lower byte is displayed. The address of the upper byte is equal to the address of the lower byte plus 1.



NOTES
1. DOUT BITS ARE BASED ON THE PREVIOUS 16-BIT SEQUENCE ($\overline{R}$ = 0).

Figure 11. Output Register Bit Assignments

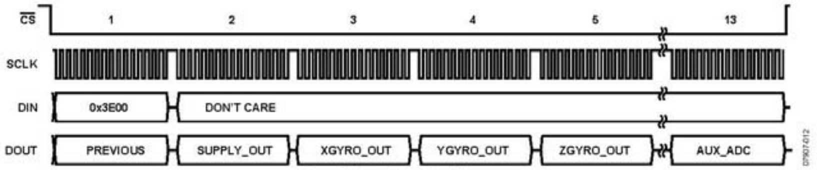


Figure 12. Burst Mode Read Sequence

OUTPUT DATA REGISTERS

Figure 6 provides the positive measurement direction for each gyroscope, accelerometer, and magnetometer. Table 9 provides the configuration and scale factor for each output data register in the ADIS16405. All inertial sensor outputs are 14 bits in length and are in twos complement format, which means that 0x0000 is equal to 0 LSB, 0x0001 is equal to +1 LSB, and 0x3FFF is equal to −1 LSB. The following is an example of how to calculate the sensor measurement from the XGYRO_OUT:

$$XGYRO\_OUT = 0x3B4A$$

$$0x000 - 0x3B4A = -0x04B6 = (4 \times 256 + 11 \times 16 + 6) - 0x04B6 = -1206 \text{ LSB}$$

$$Rate = 0.05°/\text{sec} \times (-1206) = -60.3°/\text{sec}$$

Therefore, an XGYRO_OUT output of 0x3B4A corresponds to a clockwise rotation about the z-axis (see Figure 6) of 60.3°/sec when looking at the top of the package.

Table 9. Output Data Register Formats

Register	Bits	Format	Scale
SUPPLY_OUT	14	Binary, 5 V = 0x0814	2.42 mV
XGYRO_OUT[1]	14	Twos complement	0.05°/sec
YGYRO_OUT[1]	14	Twos complement	0.05°/sec
ZGYRO_OUT[1]	14	Twos complement	0.05°/sec
XACCL_OUT	14	Twos complement	10 m*g*
YACCL_OUT	14	Twos complement	10 m*g*
ZACCL_OUT	14	Twos complement	10 m*g*
XMAGN_OUT	14	Twos complement	0.5 mgauss
YMAGN_OUT	14	Twos complement	0.5 mgauss
ZMAGN_OUT	14	Twos complement	0.5 mgauss
TEMP_OUT[2]	12	Twos complement	0.14°C
AUX_ADC	12	Binary, 1 V = 0x04D9	0.81 mV

[1] Assumes that the scaling is set to ±300°/sec. This factor scales with the range.
[2] The typical output for this register at +25°C is 0x0000.

Each output data register uses the bit assignments shown in Figure 13. The ND flag indicates that unread data resides in the output data registers. This flag clears and returns to 0 during an output register read sequence. It returns to 1 after the next internal sample updates the registers with new data. The EA flag indicates that one of the error flags in the DIAG_STAT register (see Table 23) is active (true). The remaining 14 bits are for data.

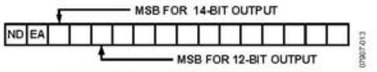


Figure 13. Output Register Bit Assignments

Auxiliary ADC

The AUX_ADC register provides access to the auxiliary ADC input channel. The ADC is a 12-bit successive approximation converter that has an equivalent input circuit to the one shown in Figure 14. The maximum input is 3.3 V. The ESD protection diodes can handle 10 mA without causing irreversible damage. The on resistance (R1) of the switch has a typical value of 100 Ω. The sampling capacitor, C2, has a typical value of 16 pF.

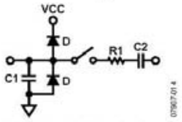


Figure 14. Equivalent Analog Input Circuit
(Conversion Phase: Switch Open, Track Phase: Switch Closed)

CALIBRATION

Manual Bias Calibration

The bias offset registers in Table 10, Table 11, and Table 12 (hard-iron correction for magnetometer) provide a manual adjustment function for the output of each sensor. For example, if XGYRO_OFF equals 0x1FF6, the XGYRO_OUT offset shifts by −10 LSB, or −0.125°/sec. The DIN command for the upper byte is DIN = 0x9B1F; for the lower byte, DIN = 0x9AF6.

Table 10. XGYRO_OFF, YGYRO_OFF, ZGYRO_OFF

Bits	Description
[15:13]	Not used.
[12:0]	Data bits. Twos complement, 0.0125°/sec per LSB. Typical adjustment range = ±50°/sec.

Table 11. XACCL_OFF, YACCL_OFF, ZACCL_OFF

Bits	Description
[15:12]	Not used.
[11:0]	Data bits. Twos complement, 3.3 m*g*/LSB. Typical adjustment range = ±6.75 *g*.

Table 12. XMAGN_HIF, YMAGN_HIF, ZMAGN_HIF

Bits	Description
[15:14]	Not used.
[13:0]	Data bits. Twos complement, 0.5 mgauss/LSB. Typical adjustment range = ±4 gauss.

Magnetometer Soft-Iron Correction (Scale Factor)

The soft-iron correction factor for the magnetometer provides opportunity to change the scale factor for each individual axis.

Table 13. XMAGN_SIF, YMAGN_SIF, ZMAGN_SIF

Bits	Description
[15:12]	Not used.
[11:0]	Data bits. Binary, linear scale adjustment factor between 0x0000 (0x) and 0x3FFF (2x).

Gyroscope Automatic Bias Null Calibration

Set GLOB_CMD[0] = 1 (DIN = 0xBE01) to execute this function, which measures the gyroscope outputs and then loads the gyroscope offset registers with the opposite values to provide a quick bias calibration. Then, all sensor data resets to 0, and the flash memory updates automatically within 50 ms (see Table 14).

Gyroscope Precision Automatic Bias Null Calibration

Set GLOB_CMD[4] = 1 (DIN = 0xBE10) to execute this function, which takes the sensor offline for 30 sec while it collects a set of gyroscope data and calculates a more accurate bias correction factor for each gyroscope. Once calculated, the correction factor loads into the three gyroscope offset registers, all sensor data resets to 0, and the flash memory updates automatically within 50 ms (see Table 14).

Restoring Factory Calibration

Set GLOB_CMD[1] = 1 (DIN = 0xBE02) to execute this function, which resets each user calibration register (see Table 10, Table 11, and Figure 12) to 0x0000, resets all sensor data to 0, and automatically updates the flash memory within 50 ms (see Table 14).

Linear Acceleration Bias Compensation (Gyroscope)

Set MSC_CTRL[7] = 1 (DIN = 0xB486) to enable correction for low frequency acceleration influences on gyroscope bias. Note that the DIN sequence also preserves the factory default condition for the data ready function (see Table 19).

OPERATIONAL CONTROL

Global Commands

The GLOB_CMD register provides trigger bits for several useful functions. Setting the assigned bit to 1 starts each operation, and the bit returns to 0 after completion. For example, set GLOB_CMD[7] = 1 (DIN = 0xBE80) to execute a software reset, which stops the sensor operation and runs the device through its start-up sequence. This includes loading the control registers with their respective flash memory locations prior to producing new data. Reading the GLOB_CMD registers (DIN = 0x3E00) starts the burst mode read sequence.

Table 14. GLOB_CMD

Bits	Description
[15:8]	Not used
[7]	Software reset command
[6:5]	Not used
[4]	Precision autonull command
[3]	Flash update command
[2]	Auxiliary DAC data latch
[1]	Factory calibration restore command
[0]	Autonull command

Internal Sample Rate

The ADIS16405 performs best when the sample rate is set to the factory default setting of 819.2 SPS. For applications that require lower sample rates, the SMPL_PRD register controls the ADIS16405 internal sample (see Table 15), and the following relationship produces the sample rate:

$$t_S = t_B \times N_S + 1$$

Table 15. SMPL_PRD

Bits	Description
[15:8]	Not used
[7]	Time base (t_B) 0 = 0.61035 ms, 1 = 18.921 ms
[6:0]	Increment setting (N_S) Internal sample period = $t_S = t_B \times N_S + 1$

For example, set SMPL_PRD[7:0] = 0x0A (DIN = 0xB60A) for an internal sample period of 6.7 ms and a sample rate of 149 SPS.

Power Management

Setting SMPL_PRD ≥ 0x0A also sets the sensor in low power mode. For systems that require lower power dissipation, in-system characterization helps users quantify the associated performance trade-offs. In addition to sensor performance, this mode affects SPI data rates (see Table 2). Two sleep mode options are listed in Table 16. Set SLP_CNT[8] = 1 (DIN = 0xBB01) to start the indefinite sleep mode, which requires a $\overline{CS}$ assertion (high to low), reset, or power cycle to wake up. Set SLP_CNT[7:0] = 0x64 (DIN = 0xBA64) to put the ADIS16405 to sleep for 50 sec, as an example of the programmable sleep time option.

Table 16. SLP_CNT

Bits	Description
[15:9]	Not used
[8]	Indefinite sleep mode, set to 1
[7:0]	Programmable sleep time bits, 0.5 sec/LSB

Digital Filtering

Programmable low-pass filtering provides additional opportunity for noise reduction on the inertial sensor outputs. This filter contains two cascaded averaging filters that provide a Bartlett window, FIR filter response (see Figure 15). SENS_AVG[2:0] controls the number of taps in each averaging stage. For example, SENS_AVG[2:0] = 110 sets each stage tap to 64. The total number of taps in the filter is equal to 2N + 1.

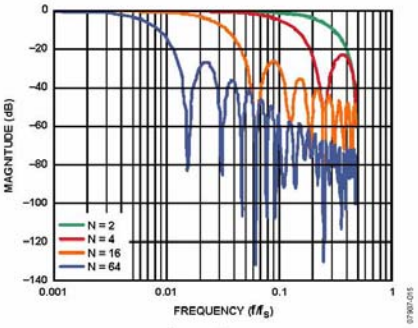


Figure 15. Bartlett Window FIR Frequency Response (Phase = N Samples)

Dynamic Range

There are three dynamic range settings for the gyroscope: ±75°/sec, ±150°/sec, and ±300°/sec. The lower dynamic range settings (±75°/sec and ±150°/sec) limit the minimum filter tap sizes to maintain the resolution as the measurement range decreases. The recommended order for programming the SENS_AVG register is upper byte (sensitivity), followed by lower byte (filtering). For example, set SENS_AVG[10:8] = 010 (DIN = 0xB902) for a measurement range of ±150°/sec, and then set SENS_AVG[2:0] = 110 (DIN = 0xB806) for 64 taps per stage (129 taps overall).

Table 17. SENS_AVG

Bits	Settings	Description
[15:11]		Not used
[10:8]		Measurement range (sensitivity) selection
	100	±300°/sec (default condition)
	010	±150°/sec, filter taps ≥ 4 (Bits[2:0] ≥ 0x02)
	001	±75°/sec, filter taps ≥ 16 (Bits[2:0] ≥ 0x04)
[7:3]		Not used
[2:0]		Number of taps in each stage $N = 2^M$

INPUT/OUTPUT FUNCTIONS

General-Purpose I/O

DIO1, DIO2, DIO3, and DIO4 are configurable, general-purpose I/O lines that serve multiple purposes according to the following control register priority: MSC_CTRL, ALM_CTRL, and GPIO_CTRL. For example, set GPIO_CTRL = 0x080C (DIN = 0xB508, and then 0xB40C) to set DIO1 and DIO2 as inputs and DIO3 and DIO4 as outputs, with DIO3 set low and DIO4 set high.

Table 18. GPIO_CTRL

Bits	Description
[15:12]	Not used
[11]	General-Purpose I/O Line 4 (DIO4) data level
[10]	General-Purpose I/O Line 3 (DIO3) data level
[9]	General-Purpose I/O Line 2 (DIO2) data level
[8]	General-Purpose I/O Line 1 (DIO1) data level
[7:4]	Not used
[3]	General-Purpose I/O Line 4 (DIO4), direction control 1 = output, 0 = input
[2]	General-Purpose I/O Line 3 (DIO3), direction control 1 = output, 0 = input
[1]	General-Purpose I/O Line 2 (DIO2), direction control 1 = output, 0 = input
[0]	General-Purpose I/O Line 1 (DIO1), direction control 1 = output, 0 = input

Input Clock Configuration

The input clock allows for external control over sampling in the ADIS16405. Set GPIO_CTRL[3] = 0 (DIN = 0x0B200) and SMPL_PRD[7:0] = 0x00 (DIN = 0xB600) to enable this function. See Table 2 and Figure 4 for timing information.

Data Ready I/O Indicator

The factory default sets DIO1 as a positive data ready indicator signal. The MSC_CTRL[2:0] register provides configuration options for changing this. For example, set MSC_CTRL[2:0] = 100 (DIN = 0xB404) to change the polarity of the data ready signal for interrupt inputs that require negative logic inputs for activation. The pulse width will be between 100 μs and 200 μs over all conditions.

Table 19. MSC_CTRL

Bits	Description
[15:12]	Not used
[11]	Memory test (clears on completion) 1 = enabled, 0 = disabled
[10]	Internal self-test enable (clears on completion) 1 = enabled, 0 = disabled
[9]	Manual self-test, negative stimulus 1 = enabled, 0 = disabled
[8]	Manual self-test, positive stimulus 1 = enabled, 0 = disabled
[7]	Linear acceleration bias compensation for gyroscopes 1 = enabled, 0 = disabled
[6]	Linear accelerometer origin alignment 1 = enabled, 0 = disabled
[5:3]	Not used
[2]	Data ready enable 1 = enabled, 0 = disabled
[1]	Data ready polarity 1 = active high, 0 = active low
[0]	Data ready line select 1 = DIO2, 0 = DIO1

Auxiliary DAC

The 12-bit AUX_DAC line can drive its output to within 5 mV of the ground reference when it is not sinking current. As the output approaches 0 V, the linearity begins to degrade (~100 LSB beginning point). As the sink current increases, the nonlinear range increases. The DAC latch command moves the values of the AUX_DAC register into the DAC input register, enabling both bytes to take effect at the same time.

Table 20. AUX_DAC

Bits	Description
[15:12]	Not used.
[11:0]	Data bits. Scale factor = 0.8059 mV/code, offset binary format, 0 V = 0 codes.

Table 21. Setting AUX_DAC = 1 V

DIN	Description
0xB0D9	AUX_DAC[7:0] = 0xD9 (217 LSB).
0xB104	AUX_DAC[15:8] = 0x04 (1024 LSB).
0xBE04	GLOB_CMD[2] = 1. Move values into the DAC input register, resulting in a 1 V output level.

DIAGNOSTICS

Self-Test

The self-test function offers the opportunity to verify the mechanical integrity of each MEMS sensor. It applies an electrostatic force to each sensor element, which results in mechanical displacement that simulates a response to actual motion. Table 1 lists the expected response for each sensor, which provides pass/fail criteria. Set MSC_CTRL[10] = 1 (DIN = 0xB504) to run the internal self-test routine, which exercises all inertial sensors, measures each response, makes pass/fail decisions, and reports them to error flags in the DIAG_STAT register. MSC_CTRL[10] resets itself to 0 after completing the routine. MSC_CTRL[9:8] (DIN = 0xB502 or 0xB501) provides manual control over the self-test function. Table 22 shows an example test flow for using this option to check the x-axis gyroscope. Zero motion provides results that are more reliable. The settings in Table 22 are flexible and provide opportunity for optimization around speed and noise influence. For example, using fewer filtering taps decreases delay times but increases the opportunity for noise influence.

Memory Test

Setting MSC_CTRL[11] = 1 (DIN = 0xB508) performs a checksum verification of the flash memory locations. The pass/fail result loads into the DIAG_STAT[6] register.

Status

The error flags provide indicator functions for common system-level issues. All of the flags clear (set to 0) after each DIAG_STAT register read cycle. If an error condition remains, the error flag returns to 1 during the next sample cycle. DIAG_STAT[1:0] does not require a read of this register to return to 0.

Table 22. Manual Self-Test Example Sequence

DIN	Description
0xB601	SMPL_PRD[7:0] = 0x01, sample rate = 819.2 SPS.
0xB904	SENS_AVG[15:8] = 0x04, gyroscope range = ±300°/sec.
0xB802	SENS_AVG[7:0] = 0x02, four-tap averaging filter.
	Delay = 50 ms.
0x0400	Read XGYRO_OUT.
0xB502	MSC_CTRL[9] = 1, gyroscope negative self-test.
	Delay = 50 ms.
0x0400	Read XGYRO_OUT.
	Calculate the positive change from the first reading to the second reading of XGYRO_OUT, and check to make sure the change is within the positive self-test response range specified in Table 1.
0xB501	MSC_CTRL[9:8] = 01, gyroscope/accelerometer positive self-test.
	Delay = 50 ms.
0x0400	Read XGYRO_OUT.
	Calculate the negative change from the first reading to the third reading of XGYRO_OUT, and check to make sure the change is within the positive self-test response range specified in Table 1
0xB500	MSC_CTRL[15:8] = 0x00.

Table 23. DIAG_STAT Bit Descriptions

Bit	Description
[15]	Z-axis accelerometer self-test failure (1 = fail, 0 = pass)
[14]	Y-axis accelerometer self-test failure (1 = fail, 0 = pass)
[13]	X-axis accelerometer self-test failure (1 = fail, 0 = pass)
[12]	X-axis gyroscope self-test failure (1 = fail, 0 = pass)
[11]	Y-axis gyroscope self-test failure (1 = fail, 0 = pass)
[10]	Z-axis gyroscope self-test failure (1 = fail, 0 = pass)
[9]	Alarm 2 status (1 = active, 0 = inactive)
[8]	Alarm 1 status (1 = active, 0 = inactive)
[7]	Not used
[6]	Flash test, checksum flag (1 = fail, 0 = pass)
[5]	Self-test diagnostic error flag (1 = fail, 0 = pass)
[4]	Sensor overrange (1 = fail, 0 = pass)
[3]	SPI communication failure (1 = fail, 0 = pass)
[2]	Flash update failure (1 = fail, 0 = pass)
[1]	Power supply above 5.25 V (1 = power supply ≥ 5.25 V, 0 = power supply ≤ 5.25 V)
[0]	Power supply below 4.75 V (1 = power supply ≤ 4.75 V, 0 = power supply ≥ 4.75 V)

Alarm Registers

The alarm function provides monitoring for two independent conditions. The ALM_CTRL register provides control inputs for data source, data filtering (prior to comparison), static comparison, dynamic rate-of-change comparison, and output indicator configurations. The ALM_MAGx registers establish the trigger threshold and polarity configurations.

Table 27 gives an example of how to configure a static alarm. The ALM_SMPLx registers provide the number of samples to use in the dynamic rate-of-change configuration. The period equals the number in the ALM_SMPLx register multiplied by the sample period time, which is established by the SMPL_PRD register. See Table 28 for an example of how to configure the sensor for this type of function.

Table 24. ALM_CTRL Bit Designations

Bits	Settings	Description
[15:12]		Alarm 2 source selection
	0000	Disable
	0001	Power supply output
	0010	X-axis gyroscope output
	0011	Y-axis gyroscope output
	0100	Z-axis gyroscope output
	0101	X-axis accelerometer output
	0110	Y-axis accelerometer output
	0111	Z-axis accelerometer output
	1000	X-axis magnetometer output
	1001	Y-axis magnetometer output
	1010	Z-axis magnetometer output
	1011	Gyroscope temperature output
	1100	Auxiliary ADC input
[11:8]		Alarm 1 source selection (same as Alarm 2)
[7]		Rate-of-change (ROC) enable for Alarm 2 1 = rate of change, 0 = static level
[6]		Rate-of-change (ROC) enable for Alarm 1 1 = rate of change, 0 = static level
[5]		Not used
[4]		Comparison data filter setting[1] 1 = filtered data, 0 = unfiltered data
[3]		Not used
[2]		Alarm output enable 1 = enabled, 0 = disabled
[1]		Alarm output polarity 1 = active high, 0 = active low
[0]		Alarm output line select 1 = DIO2, 0 = DIO1

[1] Incline outputs always use filtered data in this comparison.

Table 25. ALM_MAG1, ALM_MAG2

Bits	Description
[15]	Comparison polarity 1 = greater than, 0 = less than
[14]	Not used
[13:0]	Data bits that match the format of the trigger source selection

Table 26. ALM_SMPL1, ALM_SMPL2

Bits	Description
[15:8]	Not used
[7:0]	Data bits: number of samples (both 0x00 and 0x01 = 1)

Table 27. Alarm Configuration Example 1

DIN	Description
0xAF55, 0xAE17	ALM_CTRL = 0x5517. Alarm 1 input = XACCL_OUT. Alarm 2 input = XACCL_OUT. Static level comparison, filtered data. DIO2 output indicator, positive polarity.
0xA783, 0xA641	ALM_MAG1 = 0x8341. Alarm 1 is true if XACCL_OUT > 0.5 *g*.
0xA93C, 0xA8BF	ALM_MAG2= 0x3CBF. Alarm 2 is true if XACCL_OUT < −0.5 *g*.

Table 28. Alarm Configuration Example 2

DIN	Description
0xAF76, 0xAE87	ALM_CTRL = 0x7687. Alarm 1 input = ZACCL_OUT. Alarm 2 input = YACCL_OUT. Rate of change comparison, unfiltered data. DIO2 output indicator, positive polarity.
0xB601	SMPL_PRD = 0x0001. Sample rate = 819.2 SPS.
0xAB08	ALM_SMPL1 = 0x0008. Alarm 1 rate of change period = 9.77 ms.
0xAC50	ALM_SMPL2= 0x0050. Alarm 2 rate of change period = 97.7 ms.
0xA783, 0xA641	ALM_MAG1 = 0x8341. Alarm 1 is true if XACCL_OUT > 0.5 *g*.
0xA93C, 0xA8BE	ALM_MAG2= 0x3CBE. Alarm 2 is true if XACCL_OUT < −0.5 *g*.

OUTLINE DIMENSIONS

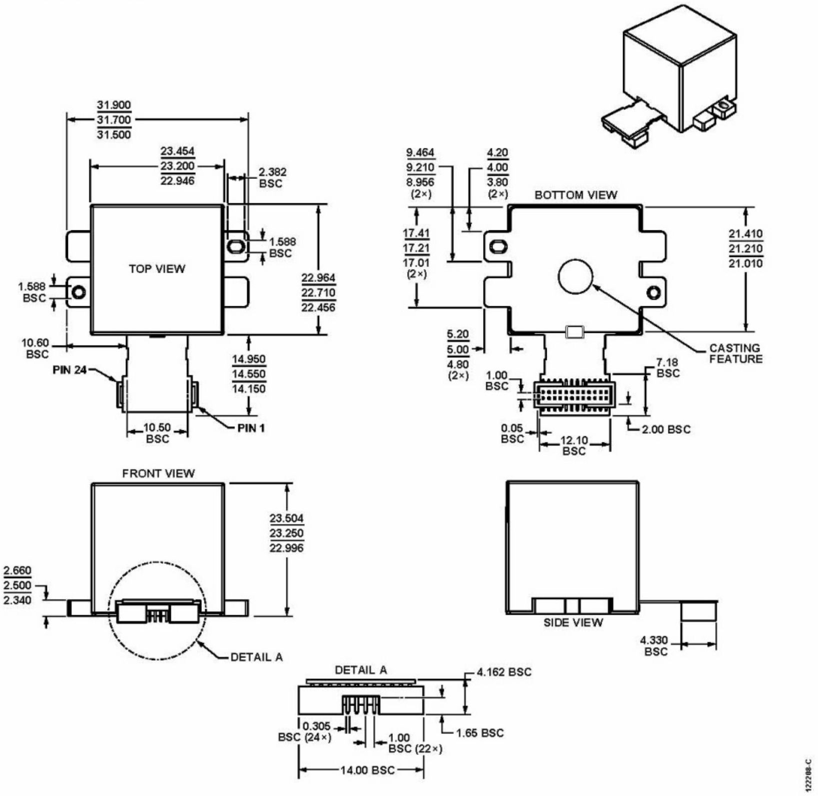


Figure 16. 24-Lead Module with Connector Interface (ML-24-2) Dimensions shown in millimeters

ORDERING GUIDE

Model	Temperature Range	Package Description	Package Option
ADIS16405BMLZ[1]	−40°C to +105°C	24-Lead Module with Connector Interface	ML-24-2
ADIS16405/PCBZ[1]		Interface Board	

[1] Z = RoHS Compliant Part.

C. SIMULINK® MODEL

1. Overall Model

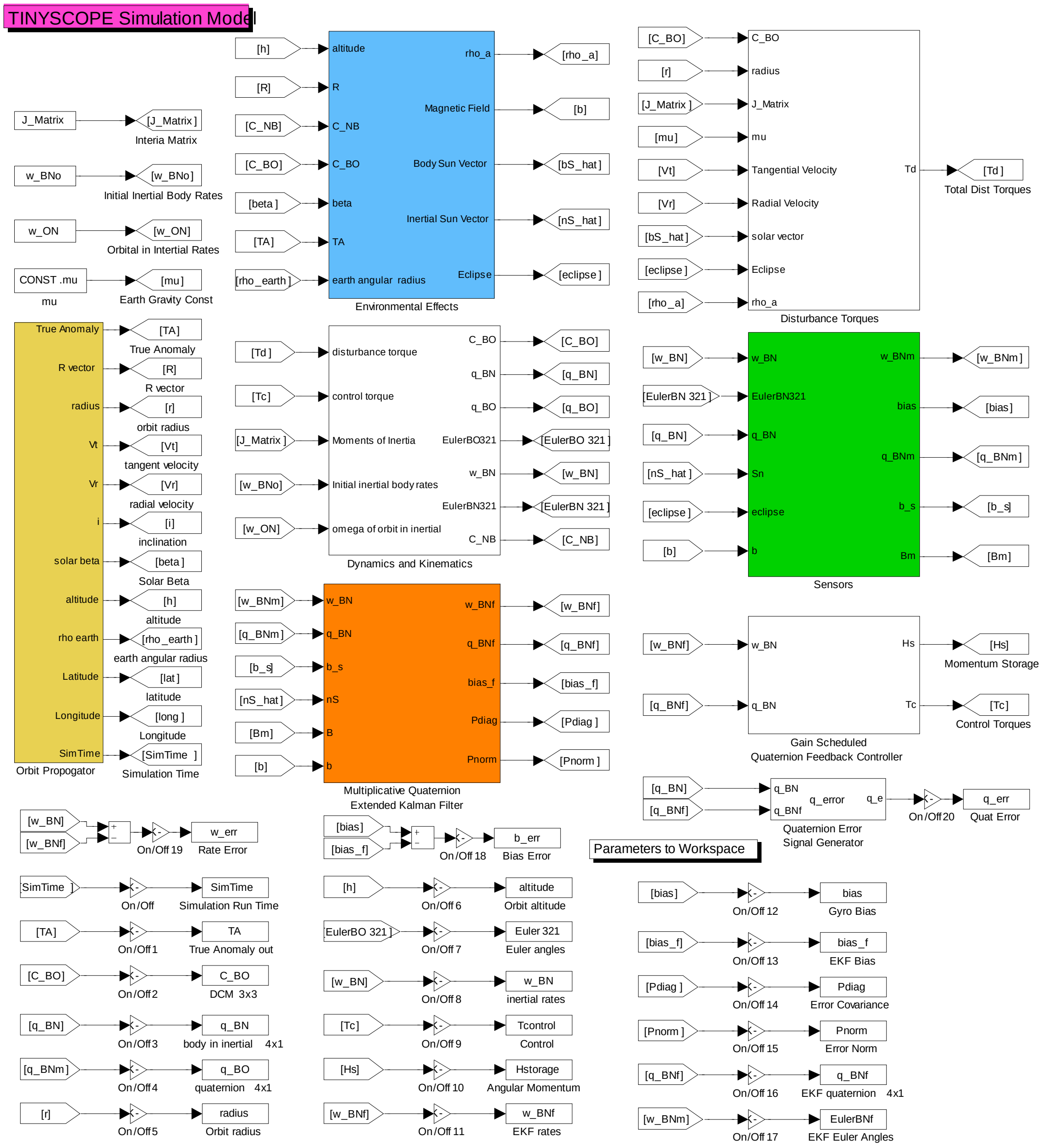


Figure 59. TINYSCOPE Overall Simulink® Model.

2. Orbital Propagator

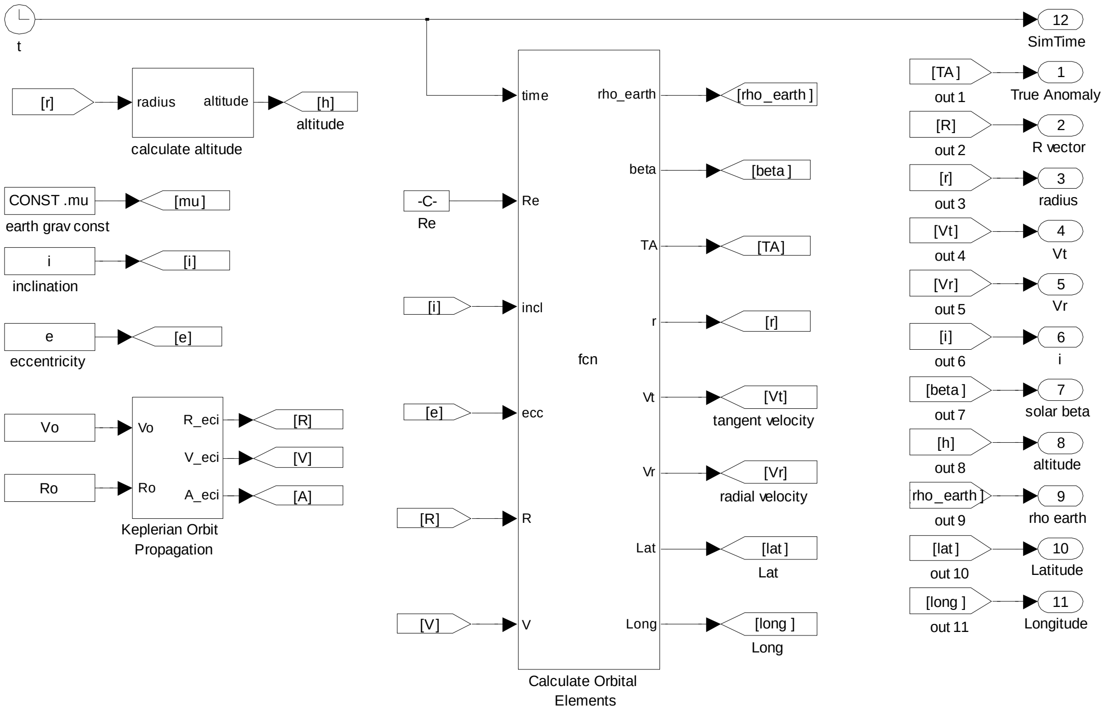


Figure 60. Orbital Propagator Block.

3. Dynamics and Kinematics

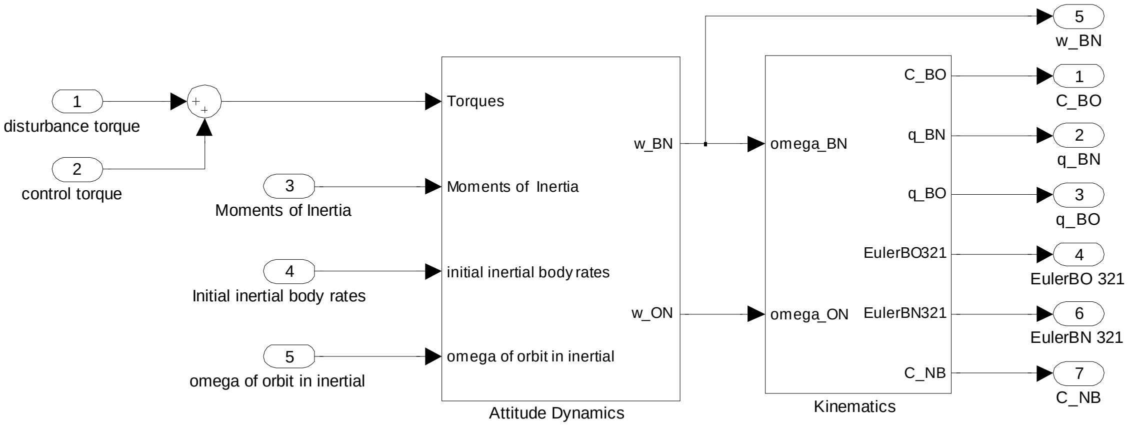


Figure 61. Spacecraft Dynamics and Kinematics Block.

4. Environmental Effects

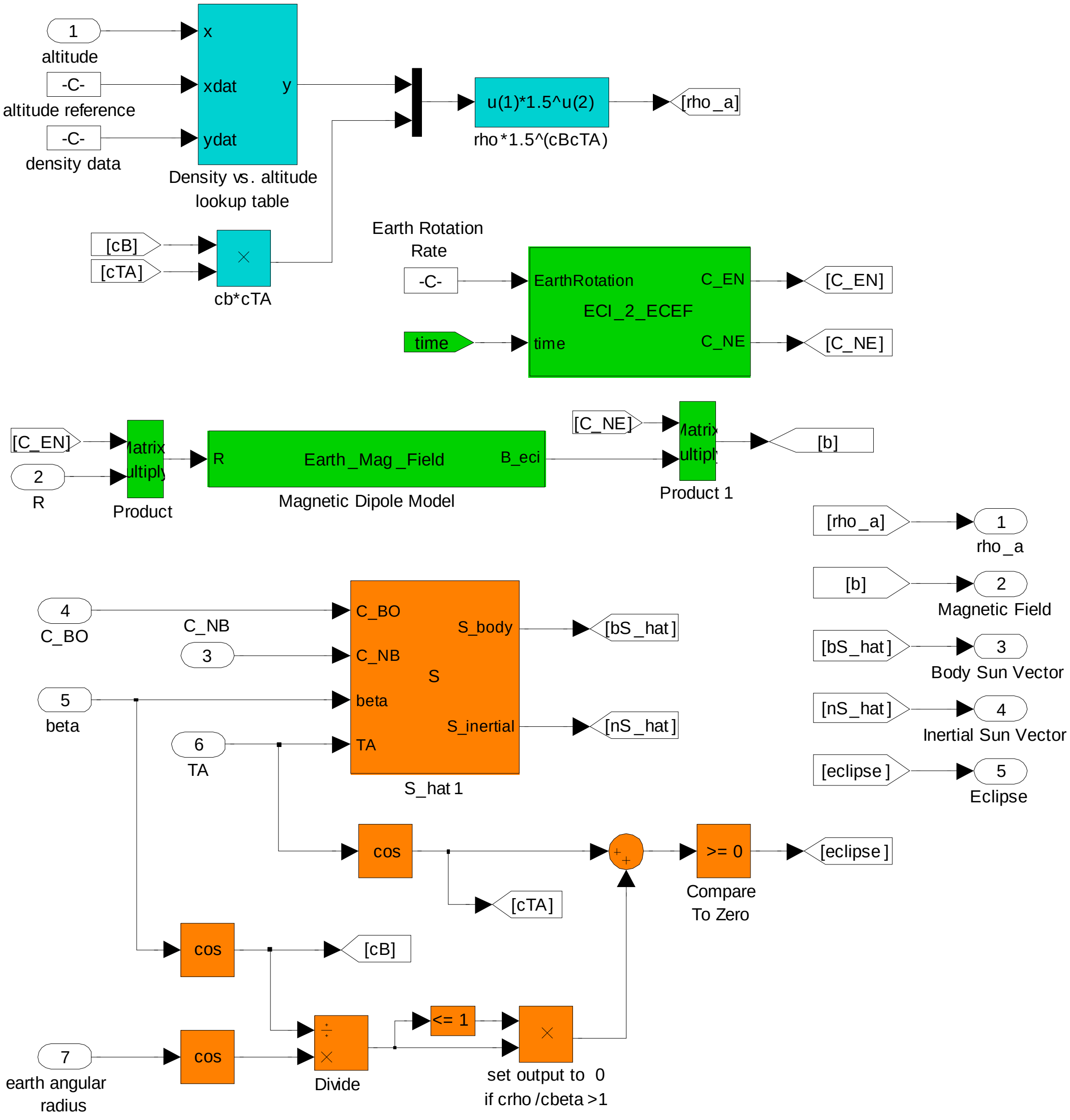


Figure 62. Environmental Effects Block.

5. Disturbance Torques

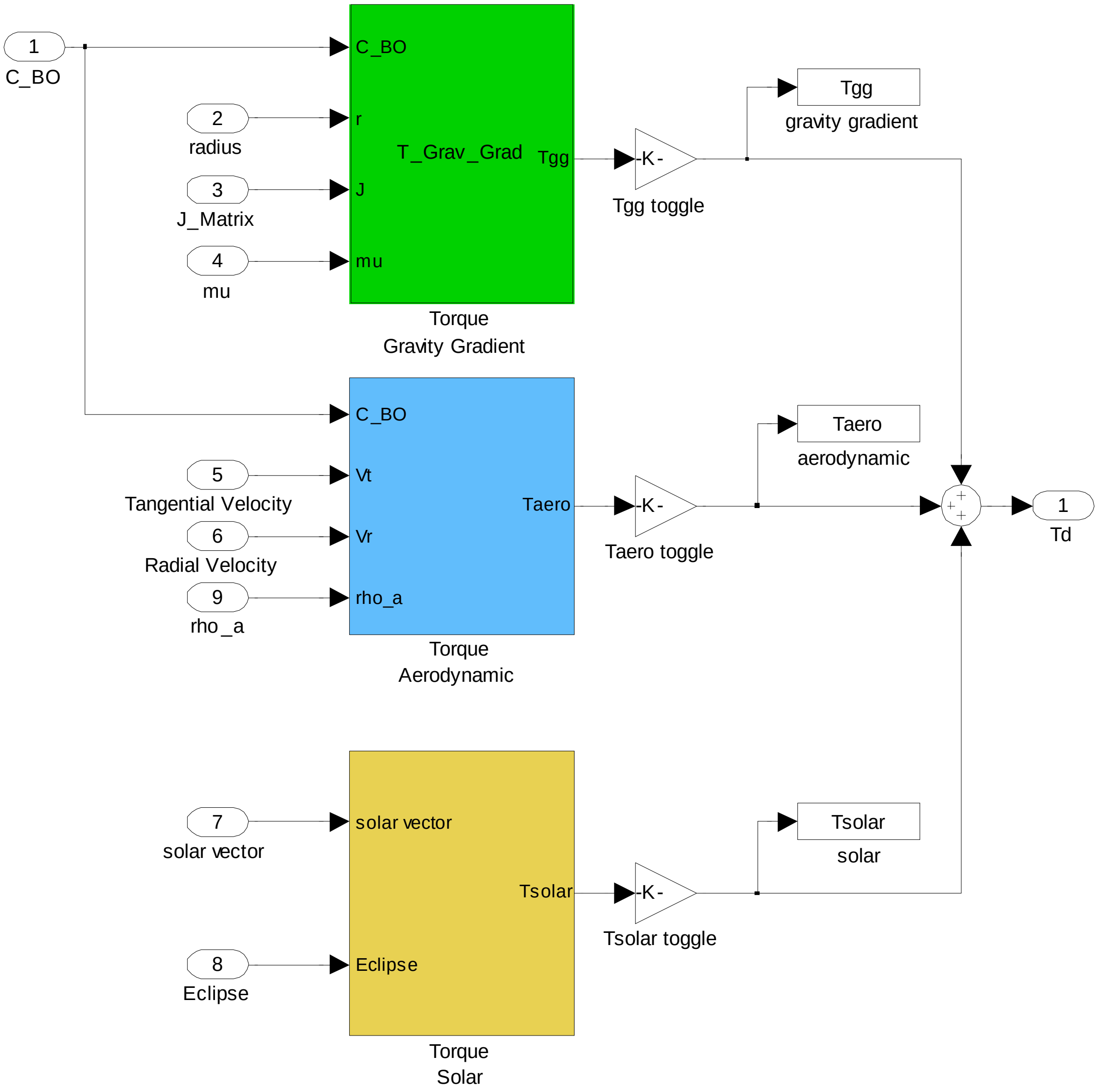


Figure 63. Disturbance Torque Block.

6. Attitude Sensors

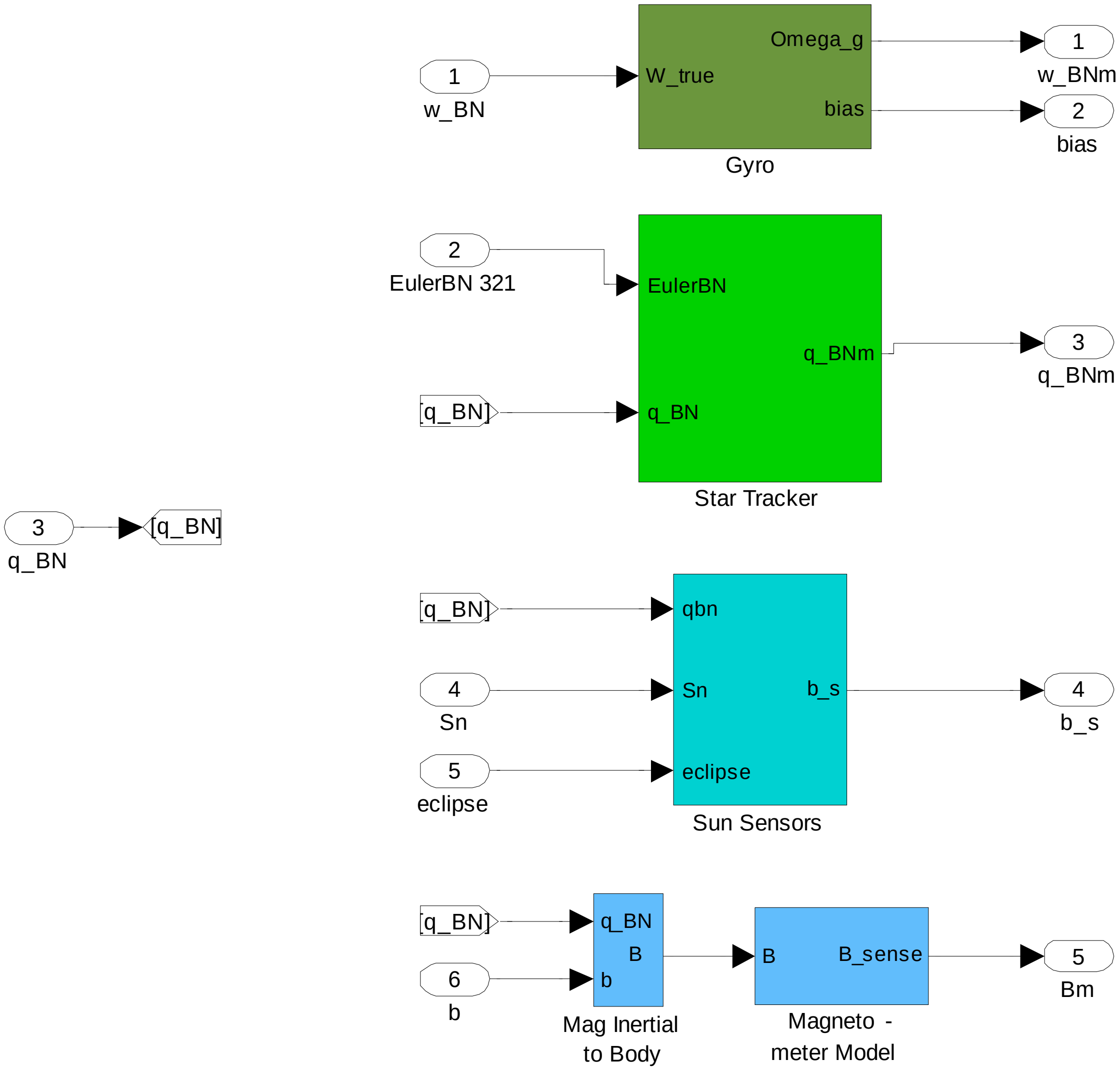


Figure 64. Simulation Attitude Sensor Block.

D. MATLAB® CODE

1. Main Script

```
%% TINYSCOPE Main Script
%
%  Author:
%  LCDR J. Allen Blocker
%  Naval Postgraduate School
%
```

```
%   LT Jason D. Tuthill
%   Naval Postgraduate School
%
%   Thesis Advisor:
%   Dr. Marcello Romano
%   Naval Postgraduate School
%
%%  Format
clear all
close all
clc

global CONST
R2D = 180/pi;
D2R = pi/180;
%%  Set Simulation Conditions

    InitialEulerN = [0 0 0];%deg
    ReferenceEuler = [0 0 0];%deg



    Kp = .25;
    Kd1 = .15;
    Kd2 = .15;
    Kd3 = .15;
    Ki = .15;

%***  Toggle switches turn the labeled functions on (1) or off (0).
***
    Tgg_toggle      = 0;%
    Taero_toggle    = 0;%
    Tsolar_toggle   = 0;%
    timeOn          = 1;
    taOn            = 0;
    cboOn           = 0;
    qbnOn           = 1;
    qbnmOn          = 1;
    rOn             = 0;
    hOn             = 0;
    e321On          = 1;
    wbnOn           = 1;
    tcOn            = 0;
    hsOn            = 0;
    wbnfOn          = 1;
    biasOn          = 1;
    biasfOn         = 1;
    pdOn            = 1;
    pnOn            = 1;
    qbnfOn          = 1;
    wbnmOn          = 1;
    werrOn          = 1;
    berrOn          = 1;
    qerrOn          = 1;
```

```
%%  Set Constants
    CONST.mu        = 398.6004418e12;%m^3/s^2
    CONST.mu_moon   = 4.902802953597e12;%m^3/s^2
    CONST.mu_sun    = 1.327122E20;%m^3/s^2
    CONST.Re        = 6.378137E6;%m                 earth radius
    CONST.Rs        = 1.4959787e11;%m               solar radius
    CONST.J2        = 1.08262668355E-3;%            J2 term
    CONST.J3        = -2.53265648533E-6;%           J3 term
    CONST.J4        = -1.61962159137E-6;%           J4 term
    CONST.SolarPress= 4.51e-6;%N/m^2                solar wind pressure
    CONST.SOLARSEC  = 806.81112382429;%TU
    CONST.w_earth   = [0;0;.0000729211585530];%r/s earth rotation
    CONST.Cd        = 2.5;%                         Coefficient of Drag
    CONST.Cr        = .6;%                          Coefficient of Re-
flect
    CONST.OmegaDot  = 1.991e-7;%rad/s               ascending node ad-
vance for sun-synch

%%  Set Orbital Elements
    %Kep elements meters and radians (a,e,i,W,w,n)

    h_p             = 500e3;%m                      altitude at perigee
    h_a             = 500e3;%m                      altitude at apogee



    RAAN = 0;%rad                                   Right Ascention
    w = 0;%rad                                      argument of perigee
    TAo = 0;%rad                                    true anomaly
    Rp  = CONST.Re+h_p;%m                           radius of perigee
    Ra  = CONST.Re+h_a;%m                           radius of apogee
    e   = (Ra-Rp)/(Ra+Rp);%(m/m)                    eccentricity
    a   = (Ra+Rp)/2;%m                              semi-major axis
    ho  = sqrt(a*CONST.mu*(1-e^2));%m^2/s           initial angular mo-
mentum
    P   = 2*pi*sqrt(a^3/CONST.mu);%sec              Orbit Period
    i_sunsynch = acosd((CONST.OmegaDot*(1-e^2)^2*a^(7/2))...
        /(-3/2*sqrt(CONST.mu)*CONST.J2*CONST.Re^2));%eqn 4.47 from Cur-
tis
    i   = i_sunsynch*D2R;%deg (rad)                 orbit inclination

    [Ro,Vo] = sv_from_coe(CONST.mu,[ho e RAAN i w TAo]);%    initial
orbital state vector

%%  Set ICs
ON = DCM(1,-90)*DCM(3,TAo+90)*DCM(1,i*R2D);
InitialEulerO = DCM2Eul(ON);
w_BNo = [0;-2*pi/P;0];%rad   initial body rates
w_ON =  [0;-2*pi/P;0];%rad

%   Sensor parameters
%   Gyro
GYRO_Bias = (3*randn(3,1))*pi/180;  % + 3 deg(rad)/sec
N_ARW = (0.029)*pi/180;
K_RRW = (0.0002)*pi/180;
```

```
ARW = N_ARW^2;                        % angular white noise Variance
RRW = K_RRW^2/3;                      % bias variance
Gg = eye(3).*(-0.01+0.02*rand(3)) +...
    (ones(3,3)-eye(3)).*(-0.0006+0.0012*rand(3)); %percent

%   Magnetometer
sigMag = 1.25e-7;
%Mag_bias = (4*randn(3,1))*1e-7;     % Tesla, +/- 4 mguass
Gm = eye(3).*(-0.02+0.04*rand(3)) +...
    (ones(3,3)-eye(3)).*(-0.0028+0.0056*rand(3)); %percent

%   Sun Sensor
S1 = [0 20 0]'*pi/180;
S2 = [0 160 0]'*pi/180;
%Gss = eye(2).*(-0.02+0.04*rand(2)) +...
%    (ones(2,2)-eye(3)).*(-0.0028+0.0056*rand(2)); %percent
FOV = 0.7;
sigSS = 0.1;
J = Bessel(sigSS/3,FOV).*pi/180;

%   Star Tracker
sigST = 70 /3 /60 /60*pi/180;       %arcsec to rad (3sig)
%Gst = eye(3).*(-0.02+0.04*rand(3)) +...
%    (ones(3,3)-eye(3)).*(-0.0028+0.0056*rand(3)); %percent

%   Kalman Filter
dt = 0.05;                            %sec (20 Hz) model speed
t_ekf = dt;                           %sec (20 Hz) ekf speed
sig(1) = sqrt(ARW);                   %rad/sec^(1/2), ARW
sig(2) = sqrt(RRW);                   %rad/sec^(3/2), RRW
sig(3) = sigST;                       %rad, Star Tracker Error
sig(4) = sigSS*pi/180;                %rad, Sun Sensor Error
sig(5) = sigMag;                      %tesla, magnetometer error

ReferenceOmega = w_ON;

[qBNo] = Euler_to_Quaternion(InitialEulerN);
[qBOo] = Euler_to_Quaternion(InitialEulerO);
[ReferenceQuaternion] = Euler_to_Quaternion(ReferenceEuler);



%%  Run Simulation
[Spacecraft]= Calculate_6U_Spacecraft;

J_Matrix = Spacecraft.MOI;

[density_table] = GetDensity;

RunTime = P;%sec

tic
sim('TinyscopeMainModel',RunTime);
```

```
Total_Model_time = toc
factor = RunTime/Total_Model_time

DisturbanceTorques.Tgg = Tgg;
DisturbanceTorques.Taero = Taero;
DisturbanceTorques.Tsolar = Tsolar;

SensorMeasurements.ST = q_BNm;
SensorMeasurements.Gyro = w_BNm;
SensorMeasurements.bias = bias;
SensorMeasurements.SS1 = ss1;
SensorMeasurements.SS2 = ss2;
SensorMeasurements.Mag = squeeze(Bm)';

EKFerror.bias = squeeze(b_err);
EKFerror.rate = squeeze(w_err);
EKFerror.quat = q_err;
EKFerror.cov = Pdiag;
EKFerror.Pnorm = Pnorm;

FilterEst.Q = squeeze(q_BNf)';
FilterEst.Gyro = squeeze(w_BNf)';
FilterEst.bias = squeeze(bias_f)';

%%  Output Results

%  PlotOrbit(R);
%
%  PlotEKFerrors(EKFerror,SimTime);
%
%  PlotMeasurements(SensorMeasurements,SimTime);

%  PlotFilter(FilterEst,SimTime);

%  PlotModelStats(w_BNs,Bs,SimTime);

%  PlotQuaternion (squeeze(q_BNf)',q_BN,SimTime);

%  PlotDisturbanceTorques (DisturbanceTorques,SimTime);

%  PlotControlTorques (Tcontrol,SimTime);

%  PlotTorques (DisturbanceTorques,Tcontrol,SimTime);

%  PlotEulerAngles (Euler321,SimTime);
%
%  PlotQuatError (q_err, SimTime);
```

2. Attitude Matrix

```
%#eml
function att = ATT( quat )
%% Making Attitude Matrix
att = transpose(XI(quat)) * PSI(quat);
return
%----------------------------------------------------------------------
```

3. XI

```
%#eml
function xi = XI( quat )
%% Making Xi Matrix
xi = [ quat(4)*eye(3) + SKEW(quat(1:3,1)) ; - quat(1:3,1)' ];
return
%----------------------------------------------------------------------
```

4. PSI

```
%#eml
function psi = PSI( quat )
%% Making Psi Matrix
psi = [ quat(4)*eye(3) - SKEW(quat(1:3,1)) ; - quat(1:3,1)' ];
return
%----------------------------------------------------------------------
```

5. Skew

```
%#eml
function sk = SKEW( vec )
%   Check it is a 3x1 or 1x3 vector
if ~(( (size(vec,1) == 3) && (size(vec,2) == 1) ) || ( (size(vec,1) ==
1) && (size(vec,2) == 3) ))
    disp('not a vector');
    return
end
sk = [0      -vec(3)  vec(2);
    vec(3)   0      -vec(1);
    -vec(2)  vec(1)   0    ];
return
%----------------------------------------------------------------------
```

6. Quaternion to Euler

```
function euler = Quat2Euler( quat )
    euler=zeros(3,1);
    a = 2*(quat(4)*quat(2)-quat(3)*quat(1));
    if a > 1
        a = 1;
    elseif a < -1
        a = -1;
    end
```

```
    euler(1)=atan2( 2*(quat(1)*quat(4)+quat(2)*quat(3)), 1-
2*(quat(1)*quat(1)+quat(2)*quat(2)));
    euler(2)=asin(a);
    euler(3)=atan2( 2*(quat(3)*quat(4)+quat(1)*quat(2)), 1-
2*(quat(2)*quat(2)+quat(3)*quat(3)));
return
%----------------------------------------------------------------------
```

7. Bessel

```
function [J] = Bessel(sig_SS,FOV)
%%  This function creates the spacial noise of the Sun Sensor

r = linspace(eps,FOV);
theta = linspace(0,2*pi)';
n = length(r);
J = zeros(n,n);

for i = 1:n
    J(i,:) = besseli(rand(n,1),r*r(i))*cos(theta)*sig_SS;
end
J(:,1) = J(:,1)./norm(J(:,1));
J(1,2:n) = J(1,2:n)./norm(J(1,2:n));
return
%----------------------------------------------------------------------
```

8. Multiplicative Quaternion Extended Kalman Filter

```
function [wk1,qk1,biask1,Pk1] = EKF(wk1t,yk1,bs,S,Bk1,B,dt,sig,mflag)

sig_v = sig(1);
sig_u = sig(2);
sig_st = sig(3);
sig_ss = sig(4);
sig_mag = sig(5);

persistent qk biask wk Pk;
% Initialize States and Measurement
if isempty(qk)
    qk = [0 0 0 1]';
    biask = zeros(3,1);
    wk = wk1t;
    sig_n=sqrt(sig_st^2+sig_ss^2+sig_mag^2);
    Pk =
dt^(1/4)*sig_n^(1/2)*(sig_v^2+2*sig_u*sig_v*dt^(1/2))^(1/4)*eye(6);

    wk1=wk;
    qk1=qk;
    biask1=biask;
    Pk1=Pk;
    return;
end

MaxST = 1;
```

```
MaxSS = 3;          % Index of last sun sensor
MaxMag = 4;         % Index of last magnetometer

SSangles = [20;160];     % Sun Sensors fram angles
SSaxis = [2; 2];        % Sun Sensor rotation axis 1 = x, 2 = y, 3 = z

%% Propagation
biask1 = biask;


Skew_w = SKEW(wk);
Mag_w  = norm(wk);

psik = (sin(1/2*Mag_w*dt)/Mag_w)*wk;
Omega = [cos(1/2*Mag_w*dt)*eye(3)-SKEW(psik) psik;
        -psik'                 cos(1/2*Mag_w*dt) ];
qk1 = Omega*qk;

Phi_11 = eye(3)-Skew_w*sin(Mag_w*dt)/Mag_w + Skew_w^2*(1-
cos(Mag_w*dt))/Mag_w^2;        % 7.59b
Phi_12 = Skew_w*(1-cos(Mag_w*dt))/Mag_w^2 - eye(3)*dt -...
7.59c
    Skew_w^2*(Mag_w*dt-sin(Mag_w*dt))/Mag_w^3;
Phi_21 = zeros(3);                                                    %
7.59d
Phi_22 = eye(3);                                                   %
7.59e

Phi = [Phi_11 Phi_12; Phi_21 Phi_22];                                  %
7.59a

Gk = [-eye(3) zeros(3); zeros(3) eye(3)];
Qk = [ (sig_v^2*dt+1/3*sig_u^2*dt^3)*eye(3) -(1/2*sig_u^2*dt^2)*eye(3)
;
      -(1/2*sig_u^2*dt^2)*eye(3)             (sig_u^2*dt)*eye(3)
];

Pk1 = Phi*Pk*Phi'+Gk*Qk*Gk';

%% Update Loop  -------------------------------------------------------
----
if(sum(mflag) >= 1)
    Att = ATT(qk1);
    delX = zeros(6,1);
    for i = 1:MaxMag
% Compute H matrix for Star Tracker Measurement -----------------------
----
        if( (mflag(i) == 1) && (i <= MaxST) )

            Xi = XI(qk1);
            H = [ 1/2*Xi(1:3,:) zeros(3,3) ];

            R = sig_st^2*eye(3);
```

```
            % Gain
            K = Pk1*H'/(H*Pk1*H' + R);

            % Update
            Pk1 = (eye(6) - K*H)*Pk1;

            res = yk1(1:3,1) - qk1(1:3,1);
            delX = delX + K*(res-H*delX);

% Update for Sun Sensor Measurement ----------------------------------
----
        elseif( (mflag(i) == 1) && (i <= MaxSS) )   % to max number of
Sun Sensors

            H = [ DCM(SSaxis(i-MaxST),SSangles(i-MaxST))*SKEW(Att*S)
zeros(3,3) ];

            R = sig_ss^2*eye(3);

            % Gain
            K = (Pk1*H')/(H*Pk1*H' + R);

            % Update
            Pk1 = (eye(6) - K*H)*Pk1;

            res = bs(:,i-MaxST) - DCM(SSaxis(i-MaxST),SSangles(i-
MaxST))*Att*S;
            delX = delX + K*(res-H*delX);


%   Update for Magnetometer Measurement -------------------------------
----
        elseif( (mflag(i) == 1) && (i <= MaxMag) )   % to max number of
Magnetometers

            H = [SKEW(Att*B) zeros(3,3)];

            R = sig_mag^2*eye(3);

            % Gain
            K = (Pk1*H')/(H*Pk1*H' + R);

            % Update
            Pk1 = (eye(6) - K*H)*Pk1;

            res = Bk1 - Att*B;
            delX = delX + K*(res-H*delX);
        end
    end
    qk1 = qk1+1/2*XI(qk1)*delX(1:3,:);
    qk1 = qnormalize(qk1'*qk1,qk1);
```

```
    biask1 = biask1 + delX(4:6,:);
else

end

wk1 = wk1t - biask1;
%   Save previous values
qk = qk1;
biask = biask1;
wk = wk1;
Pk = Pk1;

return
%----------------------------------------------------------------------
----
%% Normalizing routine for quaternions
function qk1 = qnormalize(qnorm,qk1)
while (qnorm) > 1
    if qnorm > 1 + 1e-12
        qk1 = ((3 + qnorm)/(1 + 3*qnorm))*qk1;
        %   rescale quaternion to (err^3)/32
    else
        qk1 = qk1/sqrt(qnorm);
        %   renormalize quaternion
    end
    qnorm = qk1'*qk1;
end
return
%----------------------------------------------------------------------
----
%% Make Sun Sensor DCM
function A = DCM(axis,a)
A = zeros(3,3);
switch axis
    case 1
        A = [1 0 0; 0 cosd(a) sind(a); 0 -sind(a) cosd(a)];
    case 2
        A = [cosd(a) 0 -sind(a); 0 1 0; sind(a) 0 cosd(a)];
    case 3
        A = [cosd(a) sind(a) 0; -sind(a) cosd(a) 0; 0 0 1];
end
return
%----------------------------------------------------------------------
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] R. C. Olsen, *Remote Sensing from Air and Space*. Bellingham: The International Society for Optical Engineering, 2007.

[2] Alex da Silva Curiel et al., "Small Satellites For Real Operational Missions in Earth Observation," Surrey Satellite Technology Ltd, Surrey, UK, 2006.

[3] Armen Toorian, Emily Blundell, Jordi Puig Suari, and Robert Twiggs, "Cubsats as Responsive Satellites," in *AIAA 3rd Responsive Space Conference 2005*, Long Beach, 2005, pp. 1–9.

[4] Riki Munakata, Wenschel Lan, Armen Toorian, Amy Hutputanasin, and Simon Lee, "CubeSat Design Specification Rev. 12," California Polytechnic University, San Luis Obispo, Design Specification 2009.

[5] Michael's List of Cubesat Satellite Missions. (Accessed 2009, November) [Online]. http://mtech.dk/thomsen/space/cubesat.php

[6] Mathew Richard Crook, "NPS CubeSat Launcher Design, Process and Requirements," Naval Postgraduate School, Monterey, Master's Thesis 2009.

[7] Pumpkin, Inc. (Accessed 2008, December) IMI-100 ADACS. [Online]. http://www.cubesatkit.com/docs/datasheet/DS_CSK_ADACS_634-00412-A.pdf

[8] IntelliTech Microsystems, Inc. (Accessed 2009, November) IMI-200 Miniature 3-Axis ADACS Product Specification (6-19-2008). [Online]. http://www.miniadacs.com/linked/imi-200%20specification%204.1.09.pdf

[9] Space Flight Laboratory. (Accessed 2009, November) CubeSat Compact Three-Axis Attitude Actuator and Sensor Pack. [Online]. http://www.utias-sfl.net/SFL-SPC-ACS-SYS-3.3-flyer.pdf

[10] Sinclair Interplanetary. (Accessed 2009, November) Picosatellite Reaction Wheels. [Online]. http://www.sinclairinterplanetary.com/10mNm-secwheel2009a.pdf

[11] Allen J Blocker, "Tinyscope: The Feasibility of a Tactically Useful, Three-Axis Stabilized, Earth-Imaging Nano-Satellite," Naval Postgraduate School, Monterey, CA, Master's Thesis December 2008.

[12] Christopher John Ortiona, "Systems Level Engineering of Advanced Experimental Nano-Satellites," Naval Postgraduate School, Monterey, Master's Thesis 2009.

[13] Analog Devices, Inc. (Accessed 2009, July) Triaxial Inertial Sensor with Magnetometer. [Online]. http://www.analog.com/static/imported-files/data_sheets/ADIS16400_16405.pdf

[14] Julie K .Deutschmann and Itzhack Y Bar-Itzhack, "Evaluation of Attitude and Orbit Estimation Using Actual Earth Magnetic Field Data," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 3, pp. 616–623, May 2001.

[15] Doug Sinclair. (Accessed 2008, July) SS_411 Two-Axis Digital Sun Sensor. [Online]. http://www.sinclairinterplanetary.com/sunsensor2009b.pdf

[16] AeroAstro. (Accessed 2008, December) Miniature Star Tracker. [Online]. http://www.aeroastro.com/datasheets/Miniature_Star_Tracker.pdf

[17] NovAtel Inc. (Accessed 2008, December) OEMV-1 & OEMV-1G. [Online]. http://www.novatel.com/Documents/Papers/OEMV1G.pdf

[18] IEEE Aerospace and Electronic Systems Society, "IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Laser Gyros," American National Standards Institute, New York, IEEE Standard 647–2006, 2006.

[19] Lawrence C. Ng, "On The Application of Allan Variance Method for Ring Laser Gyro Performance Characterization," Lawrence Livermore National Laboratory, Internal/Limited External Report UCRL-ID-115695, 1993.

[20] M. A. Hopcroft. (2008, October) allan v1.71. Matlab M-file.

[21] W. J. Riley, "Addendum to a Test Suite For The Calculation of Time Domain," in *IEEE Frequency Control Symposium*, 1996.

[22] IEEE Aerospace and Electronic Systems Society, "IEEE Standard Specification Format Guide and Test Procedure for Coriolis Vibratory Gyros," American National Standards Institute, New York, IEEE Standard 1431–2004, 2004.

[23] Analog Devices. (Accessed 2009, July) iSensor® PC-USB Evaluation System. [Online]. http://www.analog.com/static/imported-files/eval_boards/ADISUSBZ.pdf

[24] Quang M. Lam, Nick Stamatakos, Craig Woodruf, and Sandy Ashton, "Gyro Modeling and Estimation of Its Random Noise Sources," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, TX, 2003, pp. 11–14.

[25] The Analytic Sciences Corporation, *Applied Optimal Estimation*, Arthur Gelb, Ed. Cambridge, USA: The M.I.T. Press, 1974.

[26] Greg Welch and Gary Bishop, "An Introduction to the Kalman Filter," Department of Computer Science University of North Carolina at Chapel Hill, Chapel Hil, NC, TR 95-041, 2006.

[27] John L. Crassidis and John L. Junkins, *Optimal Estimation of Dynamic Systems*, Goong Chen and Thomas J. Bridges, Eds. New York, USA: Chapman & Hall/CRC, 2004.

[28] James W. Murrel, "Precision Attitude Determination For Multimission Spacecraft," in *Proceeding of the AIAA Guidance, Navigation, and Control Conference*, Palo Alto, CA, 1978, p. 70–87.

[29] E. J. Lefferts, F L Markley, and M. D. Shuster, "Kalman Filtering for Spacecraft Attitude Estimation," in *AIAA 20th Aerospace Sciences Meeting*, Orlando, FL, 1982.

[30] Malak A. Samaan and Mohammad Abdelrahman, "Satellite Attitude Determination Utilizing Measurement Sensor and Kalman Filtering," in *ACSE 05 Conference*, Cairo, Egypt, 2005.

[31] Paul M.Oppenheimer, "Attitude Control System For Agile Nano-Satellites," Naval Postgraduate School, Monterey, CA, Master's Thesis 2009.

[32] Norman S. Nise, *Control Systems Engineering*. Hoboken: John Wiley & Sons, Inc., 2004.

[33] Gene F. Franklin, J David Powell, and Abbas Emami-Naeini, *Feedback Control of Dynamic Systems*. New York: Addison-Wesley Publishing Company, 1994.

[34] Paul Zarchan and Howard Musoff, *Fundamentals of Kalman Filtering A Practical Approach*. Reston: American Institute of Aeronautics and Astronautics, 2000.

[35] Katsuhiko Ogata, *Modern Control Engineering*. Upper Saddle River: Prentice Hall, 1997.

[36] ORS Office. (2009, January) Operationally Responsive Space Office: 2009 Naval Postgraduate School Spacecraft Design Class.

[37] James R. Wertz and Wiley J. Larson, *Space Mission Analysis and Design*. Hawthorne: Microcosm Press, 2007.

[38] Bong Wie, *Space Vehicle Dynamics and Control*. Reston: American Institute of Aeronautics and Astronautics, 2008.

[39] C. Chance Litton, "Tinyscope: The Feasibility of a Tactically Useful Earth-Imaging Nanosatellite and a Preliminary Design of the Optical Payload," Naval Postgraduate School, Monterey, CA, Master's Thesis January 2009.

[40] Scott Flagg, Rich White, and Roberta Ewart, "Operationally Responsive Space Specifications and Standards: An Approach to Converging with the Community," in *AIAA Space 2007 Conference & Exposition*, Long Beach, 2007, pp. 1–19.

[41] Les Doggrell, "Operationally Responsive Space: A Vision of the Future of Military Space," *Air & Space Power Journal*, pp. 1–9, 2006.

[42] Ryan Nugent, Riki Munakata, Alexander Chin, Roland Coelho, and Jordi Puig-Suari, "The CubeSat: The Picosatellite Standard for Research and Education," in *AIAA Space 2008*, San Diego, 2008, pp. 1–11.

[43] Howard D. Curtis, *Orbital Mechanics for Engineering Students*. Oxford, United Kingdom: Elsevier Butterworth Heinemann, 2005.

[44] Roger R. Bate, Donald D Mueller, and White E Jerry, *Fundamentals of Astrodynamics*. New York, United States of America: Dover Publications, Inc., 1971.

[45] R. C. Olsen, "Introduction to the Space Environment," Naval Postgraduate School, Monterey, United States of America, Class Guide 2005.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California