ON-BOARD ORBIT DETERMINATION AND 3-AXIS ATTITUDE DETERMINATION FOR PICOSATELLITE APPLICATIONS

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

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

by
John Arthur Bowen
July 2009

COMMITTEE MEMBERSHIP

TITLE:	On-Board Orbit Determination and 3-Axis Attitude Determination for Picosatellite Applications
AUTHOR:	John Arthur Bowen
DATE SUBMITTED:	July 2009
COMMITTEE CHAIR:	Dr. Jordi Puig-Suari
COMMITTEE MEMBER:	Dr. Kira Abercromby
COMMITTEE MEMBER:	Dr. Eric A. Mehiel
COMMITTEE MEMBER:	Dr. Marcello Romano

ABSTRACT

On-Board Orbit Determination and 3-Axis Attitude Determination for Picosatellite Applications

John Arthur Bowen

This thesis outlines an orbit determination and 3-axis attitude determination system for use on orbit as applicable to 1U CubeSats and other picosatellites. The constraints imposed by the CubeSat form factor led to the need for a simple configuration and relaxed accuracy requirements. To design a system within the tight mass, volume, and power constraints inherent to CubeSats, a balance between hardware complexity, software complexity and accuracy is sought. The proposed solution consists of a simple orbit propagator, magnetometers with a magnetic field look-up table, Sun sensors with an analytic Sun direction model, and the TRIAD method to combine vector observations into attitude information. The orbit propagator is a simple model of a circular trajectory with several frequently updated parameters and can provide orbital position data with average and maximum errors—when compared to SGP4—of less than 3.7km and 10.7km for 14 days. The magnetic field look up table provides useful information from a small memory footprint; only 480 data points provide a mean error of approximately 0.2° and a maximum error of approximately 2°—when compared to the IGRF model. The Sun's direction is modeled, and as expected, can be modeled simply and accurately. Combining the magnetic field and Sun direction models with inaccurate sensors and the TRIAD method results in useful attitude information from a very simple system. A system with Sun sensor error standard deviation of 1° and magnetometer error standard deviation of 5° yields results with average error of only 2.74°, and 99% of the errors in this case are less than approximately 13°. The system outlined provides crude attitude determination with software and hardware requirements that are well within the capabilities of current 1U CubeSats—something that many other systems, such as Kalman filters or star trackers, cannot do. It also provides an excellent starting point for future ADCS systems, which will significantly increase the ability of CubeSats.

Keywords: orbit determination, orbit propagation, attitude determination, CubeSat, picosatellite, Sun-Mag, IGRF, TRIAD.

Acknowledgments

To my friends and family, I could not have accomplished this without all of your love and support. Mom and Dad, you instilled in me ambition, a stubborn persistence, and an unquenchable thirst for knowledge, and these characteristics have served me well. I cannot express how thankful I am for everything you have done for me; without you I never could have come this far. Kirsten and Kelsey, you are both unique and beautiful people and have added valuable perspective to my life. You have taught me lessons that only a sibling could; many of the things you taught me may have gone unnoticed at the time, but in retrospect, I would not be the same person without your influences, thank you.

All of my friends throughout my career at Cal Poly, you forced me to get my head out of the books and you made that time priceless. I could never have gotten through all of the school work if I didn't have late night partners in procrastination or fun weekends—and weeknights—to look forward to. Whether it was ping pong in the dorms, beer pong on Murray Street or world-class beer 6th year, I have had an amazing time and will always look back on these times fondly.

Rebecca Payne, this thesis would not have any punctuation if it were not for your editing and advice; this thesis would not be a thesis if it had not been for your love and support. You pushed me to work harder, put up with me when I was overwhelmed, added perspective when I had lost it, and guided me through this entire process. Thank you for putting up with me during this trying endeavor, and thank you for all of the love and support along the way. It was always easier to work hard during the week if I knew I had a fun weekend with you ahead.

Dr. P, thank you for your guidance—and lack thereof; you provided invaluable direction throughout my thesis work, and you gave me plenty of rope with which to hang myself. You were the only person who truly understood where my thesis was going—including myself—and your guidance on the scope of my work made it into a coherent work of finite length. You also let me work on what interested me in the way that I wanted to; I taught myself many lessons by following the wrong path and losing perspective on my work. Thank you for the guidance when I needed it and the liberty to work on what I wanted and make some mistakes.

The present and past members of the PolySat/CubeSat lab, I have learned more in the lab than in my entire 6 years of classroom education and this is all thanks to you, my peers. Thank you to the early members who sowed the seeds of this program so that we could work in one of the most exciting labs in the world. Thank you to everyone in the lab who has entertained my questions about subjects I have no business understanding; working with everyone in the lab has been both fun and enlightening. I am confident that my experience in this lab will be valuable throughout my career, thank you for making it possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

a	=	Magnetic reference radius (6371.2 km)
A	=	Amplitude with respect to trigonometry, rotation with respect to attitude
δ_i^j	=	Kronecker delta (for *i* and *j*)
f	=	Frequency
g_n^m, h_n^m	=	IGRF Gaussian coefficients
$\vec{h}$	=	Orbital angular velocity vector
i	=	Inclination
I, J, K	=	1,2, and 3 axes in the ECEF and ECEI coordinate frames
n	=	Mean motion
u	=	Argument of latitude
ν	=	True anomaly
ω	=	Angular velocity
Ω	=	Right ascension of the ascending node
p	=	Semilatus rectum
$P_n^m(\theta)$	=	Schmidt normalized associated Legendre functions
$P^{n,m}(\theta)$	=	Gauss normalized associated Legendre functions
ϕ	=	Longitude (East positive)
$\hat{r}$	=	Reference triad unit vector
R	=	Radius
$\hat{s}$	=	Observed triad unit vector
θ	=	Coelevation (also used as a generic angle)
$\hat{V}$	=	Reference unit vector

Chapter 1: Introduction

1.1 Background

1.1.1 CubeSat

The CubeSat standard was developed through collaboration between Dr. Jordi Puig-Suari of California Polytechnic State University San Luis Obispo (Cal Poly) and Dr. Bob Twiggs of Stanford University in 1999. This standard requires that the satellites are 10cm cubes, less than 1 kg, electronically off while in the launch vehicle, and meet interfacing requirements. The standard was developed to provide universities frequent and cost effective access to space by creating a simple standard which allows satellite developers to interface with the Poly Picosatellite Orbital Deployer (P-POD). The P-POD is a flight proven satellite deployment mechanism, developed at Cal Poly, which can be interfaced with any launch vehicle. CubeSats are typically launched as secondary payloads; this is possible because of standardization, rigorous testing protocol and the use of flight proven technology. With almost 30 CubeSats in orbit and over one hundred teams actively designing CubeSats, the CubeSat concept has been greatly successful in providing access to space.

1.1.2 PolySat

PolySat is the multidisciplinary CubeSat development team at Cal Poly, which was created in 1999. The mission of the PolySat lab is to educate and to advance spaceflight through the design, construction, test, launch and operation of picosatellites. The PolySat team has designed several CubeSats: CP1 and CP2 were lost in the DNEPR-1 launch failure, CP3, CP4 and CP6 are currently on orbit and operational, and CP5 and CP7 are in the development phases. The CPX bus has been used since CP2 and has been refined continually. The goal of the CPX bus is to provide

a reliable satellite bus, including power management, command and data handling, and communication, while reserving mass, volume and power for a payload.

1.2 Motivation

The CubeSat community works within very tight mass and volume constraints, which impact the abilities of CubeSats in many ways. Two of the biggest limitations of CubeSats are communication bandwidth and power. Both of these limitations are tied to the fact that most CubeSats are tumbling. Without the ability to point the satellite, the communications antennas and the solar cells cannot be pointed. This typically leads to a configuration with a low-gain omni-directional antenna and body mounted solar cells. The addition of onboard attitude determination and control has the potential to increase CubeSat capabilities in both of these areas, significantly increasing the scope of CubeSat missions.

Many CubeSats have used passive attitude stabilization, and some have proposed the use of spin stabilization or other 1-axis techniques. These solutions are very useful for particular missions—and have been very cleverly used—but cannot meet the attitude determination and control requirments of all missions. Historically, spacecraft advanced from no control to 1-axis control and eventually 3-axis control; this process will eventually happen with CubeSats as well. Missions exist which demand 3-axis control, and 3-axis control will certainly be acheived in CubeSats as the technology implemented in them advances.

Several solutions for achieving 3-axis control have been suggested in the CubeSat community; however, none of these solutions are viable as bus implementations on a 1U CubeSat with current technology.

Star trackers are a common solution for determining spacecraft attitude in modern spacecraft. Star trackers can provide extremely accurate attitude information—less than 0.01° error[1]—while requiring a relatively small portion of a spacecraft's mass, volume and power. There has been a

push to miniaturize star sensors for nanosatellite or picosatellite use, with the idea that compromises can be made to provide a system that will fit within the requirements of these small satellites while providing accuracy on the order of degrees. The systems demonstrated to date may work for a nanosatellite mission. However, the suggested star sensor systems are nearly the size of a complete 1U CubeSat payload and require more power than a CubeSat can produce; therefore, these sensors are not a viable option for current 1U CubeSat missions requiring continuous AD from the satellite bus. These star sensors could be used for certain missions where the star sensor is the payload, like algorithm testing, but this is not the focus of this work.

Another commonly proposed solution is to take a single vector observation and use a filtering algorithm to deduce 3-axis attitude information. It has been shown[2] that filtering methods like Extended Kalman Filters can theoretically produce 3-axis attitude knowledge from single vector observation such as a magnetic field observation or a Sun direction observation. This approach pushes all the complexity to software, which is elegant in that: it requires only one sensor, is potentially tolerant of noisy or inaccurate data, and can provide other useful information like moments of inertia[2]. However, these solutions have their limitations: the filters must be very finely tuned to ensure convergence and a second attitude observation may be needed to confirm convergence. The low accuracy and noise associated with some COTS sensors used in picosatellites can make convergence inconsistent. Convergence can also take many orbits, which means the filter may need to run continuously if attitude data is required frequently. Also, the processing power available on picosatellites is usually extremely low because of the power limitations, which in turn limits software complexity. The computational complexity of these algorithms makes their use questionable for picosatellites with currently available low power processors.

Another issue is that sensors like magnetometers and horizon sensors require orbital position knowledge. There are two commonly proposed solutions for finding orbital position, GPS and

orbit propagation with propagators like SGP4 (Simplified General Perturbations Satellite Orbit Model 4), but neither of these are well suited for picosatellite applications. Current GPS technology is too large, heavy, and power hungry for implementation in a picosatellite bus. SGP4 and other similar propagators require computations that are outside of the scope of current low power processors available to picosatellites.

A compromise between the above attitude determination and orbit determination solutions is needed. In the following work I will propose a system that balances hardware and software complexity while sacrificing some accuracy to achieve a system design that can be implemented as a bus addition on a picosatellite.

1.2.1 Potential Gains Afforded by AD&C

Attitude control has the potential to significantly increase the capabilities of CubeSats by increasing the power available through the use of pointed solar arrays. Wertz[1] suggests that pointed solar arrays are roughly four times as efficient as body mounted solar arrays. Solar cell efficiency is a function of incidence angle[3] and is often approximated by:

$$efficiency = \left(\cos\left(incidence\ angle\right)\right)^3$$

the "cosine cubed" approximation. Testing of the solar cells used for PolySat satellites confirms that the cosine cubed approximation is relatively accurate. These two assumptions, along with simple math, yield some interesting design points. If a solar array—of area equal to that of the total area of body mounted panels—can be pointed to within 50°, then the worst case efficiency would be equal to that of body mounted panels. In a similar scenario, if the panel can be pointed to within 37°, then the worst case efficiency is 200% as good as that of the body mounted case. Realize also that these "worst case" scenarios are much worse than the actual case, since the assumption is that the incidence angle on the panel is always 50° or 37°, when in reality it would often be much better.

As mentioned previously, communication bandwidth is one of the areas where CubeSats are most limited. Directional antennas provide much higher gain, and are therefore much more ideal; however, directional antennas require attitude control to keep the antenna pointed at a target. The GeneSat-1 spacecraft is a good example of a CubeSat—although a 3U—which had a directional antenna. GeneSat-1 used passive magnetic control along with a patch antenna. By aligning the spacecraft with the Earth's magnetic field, the GeneSat-1 team was able to design an antenna that would have a maximum pointing error of 45° for their ground station location. Deployable high gain antennas are conceivable if attitude control and the requisite attitude determination can be provided.

Many payloads also need pointing or attitude knowledge. Even crude attitude control could be useful to many payloads. To point an Earth imaging payload generally at Earth requires only crude pointing, because in LEO the Earth occludes between 65° and 70°. To point a star sensor at a dark portion of the sky requires even less pointing accuracy, since all that is needed is to avoid the Earth, Sun, and Moon. Other payloads have various pointing requirements, but most CubeSat payloads are not extremely demanding, since they come from a design space which traditionally has no pointing capability.

1.3 Approach to Orbit Determination and Attitude Determination for Picosatellites

To achieve continuous 3-axis attitude determination in a picosatellite bus, a balance between software and hardware complexity must be made, along with a compromise in accuracy.

To eliminate the need for complex filtering routines like extended Kalman filters—which are required when only one vector observation is available—two vector observations will be used. There are multiple sensors which can provide vector observations on picosatellites with very little of the satellite's precious resources.

Magnetometers have flown on many CubeSats, and commercial off the shelf (COTS) magnetometers are small, inexpensive, reliable, and require very little power. In fact, COTS magnetometers are so inexpensive—in many ways—to fly that the CPX bus carries 5 2-axis magnetometers. This affords redundancy and increased accuracy through averaging, which in turn allows the use of relatively crude sensors. Good results have been found on the CPX spacecraft, and in the future, even better results could be achieved with a few simple processes. Accuracy could be gained if the magnetic field of the satellite is thoroughly characterized and then magnetic field readings are corrected accordingly. Deployables are to be avoided at all costs, but magnetometers on booms provide very good information and historically have been one area where a deployable is worthwhile. A small deployable boom could provide even better magnetometer data. Clearly the possibilities with magnetometers on picosatellites are promising.

Many people in the CubeSat community have proposed using solar cells as crude Sun sensors. Orbital data has proven that this is to be used only as a check and not as a vector observation. However, it has been shown that COTS photodiodes can be used as Sun sensors. Like COTS magnetometers, these sensors are small, inexpensive and require very little power. Obviously, Sun sensors can only provide attitude knowledge when in the Sun (roughly two thirds of the time for LEO spacecraft). However, attitude knowledge is most useful in the Sun for many reasons. Pointing solar arrays is only useful when in the Sun. Also, most missions would utilize attitude determination while the spacecraft is lit because attitude control, high bandwidth communication, and many payloads require a large portion of the spacecraft's power. Power can be stored in batteries—and usually is—but a concept of operation that pushes the majority of power use to the lit portions of the orbit is more efficient.

Very little research has been documented on the application of Earth observation based sensors—like Earth sensors, horizon sensors, and horizon trackers—for picosatellite applications. While it is not the focus of this research, some initial conceptual design was performed, and it is the

opinion of this author that a relatively simple, solid state Earth sensor could provide useful attitude knowledge. The limited research into this area showed that a Sun sensor is a better choice if attitude knowledge is only needed for the lighted portion of the orbit, but that if attitude knowledge is needed in eclipse, then Earth sensors may be a viable solution.

GPS, star sensors (which yield inertial measurements unlike the other sensors mentioned previously), and commonly available orbit propagators are not viable options; therefore, an orbit determination solution is required. A compromise between orbital position knowledge accuracy and computational complexity is needed. One of the simplest possible orbit propagators is a model of the orbit represented by angular velocity, time, and a circular path. In the following sections I will show that this method—along with some correction factors—can be used to gain orbital position knowledge accurate enough to provide significant gains in many CubeSat missions.

1.4 Concept of Operations

The approach outlined in the previous section suggests orbit determination through a simple orbit propagator and 3-axis attitude determination from two vector observations. Several things need to happen to accomplish orbit determination and attitude determination (ODAD) on-board. The spacecraft must have knowledge of where the observed vectors are expected to be in some non-body frame so that it can compare sensor readings to something. The spacecraft must have a routine to extract attitude information out of four vectors—two in the body frame and two in some other reference frame. In the case of Earth referencing sensors—like magnetometers—the spacecraft must have real-time knowledge of its position. Orbit determination will be accomplished with a simplified orbit propagator, which is defined by several parameters; propagator parameters must be kept up to date. Current time will also be needed for several calculations: the location of potential reference bodies—like the Sun—can be expressed as a

simple function of time, coordinate transformations require current time, and the orbit propagator requires time to define orbital position.

Two inputs are required to maintain the attitude determination and orbit determination system, namely, the orbit propagation parameters must be updated and the on-board clock must be kept accurate. This will be accomplished through uplink from the operating ground station. Orbit propagator parameters will be optimized—from the most current TLE—before attitude determination is needed. These parameters will be uplinked to the spacecraft along with the command to run the ODAD routine. The satellite can be told to run the routine until told otherwise, run immediately and stop at some future time, or to run in some future window. The on-board clock will also be reset before the routine is to be run. For a mission that demands continuous attitude determination, these commands could begin every pass. It is typical to check the health status of a CubeSat at the start of communication; the command that requests health status could be replaced with a command that updates the ODAD routine, and requests health status. The beginning of every pass would consist of updating the ODAD system, checking the spacecraft health, and then proceeding with the mission.

1.5 Requirements

Requirements for a useful attitude determination system can be set based on the limitations inherent in CubeSat development, and the potential gains associated with attitude determination and control. For an attitude determination and control (AD&C) system to be useful to a payload, it must provide accurate AD&C while not encroaching on the resources available to the payload—including mass, volume and power.

The total mass available for a 1U CubeSat is 1000 grams; typically half of this is available to the payload in the CPX bus. To be a reasonable bus implementation, the additional components for attitude determination and orbit determination need to be less than 100 grams, ideally 25.

The total volume available for a 1U CubeSat is approximately 1000cm^3; the payload volume is approximately 90x85x60mm—roughly 450cm^3—in the CPX bus. To avoid encroaching on the 550cm^3 available for the bus, an ODAD system must not be more than 100 cm^3—ideally 25 cm^3.

The electronic power system (EPS) of the CPX bus consists of body mounted solar cells and lithium ion secondary batteries; this system is capable of continuously supplying approximately 1 Watt. Attitude determination and control may very well provide increased power by enabling the use of tracking solar arrays. However, this may not be the case for initial tests, attitude determination and control system (ADCS) startup, or a lost in space scenario; therefore, power available is assumed to be 1 Watt. The large majority of this power is for use by the payload; the communication system also takes a significant portion when transmitting. The ODAD system may only need to operate while payload tests are happening; however, this is typically the state when the power budget is most negative. Therefore, power usage should be kept to a minimum. Average power usage of 50mW is a significant fraction of the bus power and will serve as a hard maximum; 25mW is a realistic value for running one low-power processor and is the target for this research.

The mass, volume and power requirements outlined above essentially limit an ODAD system to small surface mount sensors and a small printed circuit board dedicated to data acquisition and processing.

Perturbations to orbits make orbit propagation dynamic, and necessitate constant update of the parameters which define the orbit. The parameters which define the orbit will need to be uplinked to the spacecraft for on-board orbit propagation. Low Earth orbits typically provide 2-6 passes for a particular ground station per day; however, experience with CP3/CP4 has shown that consistent uplink is not always a reality. For this reason, uplink frequency—how frequently we uplink, not the inverse of period—is a driving requirement for on-board orbit propagation. Also, the orbit

propagator cannot dominate the uplink, since the majority of the uplink bandwidth must be reserved for payload operations. The orbit propagator must stay accurate for at least 2 days without update, ideally 8 days.

As outlined in Section 1.2.1, attitude control accurate to 60° may be useful to CubeSats; attitude control with error less than 30° would clearly be useful. A formal study is required to determine the breakdown of errors associated with different attitude determination and control systems; however, it is reasonable to assume the systems will have errors of the same order of magnitude. A thorough study of control systems applicable to picosatellites is not within the scope of this thesis; therefore, it will be assumed that the error in a CubeSat attitude determination and control system is approximately evenly divided between determination errors and control errors. This yields approximate maximum attitude determination error requirements of 30° and 15° for the maximum and target respectively.

Table 1 summarizes the requirements for this study. The "mandatory" column describes the values that are required for an on-board ODAD system to be viable—as a bus implementation—for 1U CubeSat missions. The "target" column contains values that define a system which would be very useful while not encroaching on the payload.

Table 1 Requirements

Requirement	Mandatory	Target
Maximum mass [grams]	100	25
Maximum volume [cubic centimeters]	100	25
Maximum average power usage[milliwatts]	50	25
Maximum OD parameter uplink frequency [times per day]	0.5	0.125
Maximum attitude determination error [degrees]	30	15

1.6 Coordinate Frames

A variety of coordinate frames are used in this work. They are briefly explained below to avoid any confusion.

1.6.1 Earth Centered Equatorial Fixed (ECEF)

Origin: Center of Earth

Fundamental Plane: Earth's equator

Principal direction: 0° Latitude 0° longitude (towards the prime meridian, in the equatorial plane)

This coordinate system is in Cartesian coordinates when referred to as ECEF. Axis will commonly be referred to as $\hat{I}, \hat{J}$, and $\hat{K}$, where $\hat{I}$ is the principal direction, $\hat{K}$ is north facing and normal to the fundamental plane and $\hat{J}$ lies in the fundamental plane completing the right orthonormal triad. The Prime Meridian—representing the zero longitude band—is the longitude band that passes through the Royal Observatory in Greenwich, United Kingdom.

1.6.2 Geocentric Latitude, Longitude, Radius (LLR)

This is the same coordinate system as ECEF expressed in spherical coordinates instead of Cartesian coordinates. Latitude is the angular measurement North and South of the Equator, North Positive. Longitude is the angular measurement East and West of the Prime Meridian, East positive. Converting between LLR and ECEF is a simple matter of Cartesian versus spherical geometry.

Colatitude and coelevation are also used in this coordinate frame instead of latitude; they represent the complementary angle to latitude.

1.6.3 Earth Centered Equatorial Inertial (ECEI)

Origin: Center of Earth

Fundamental Plane: Earth's equator

Principal direction: Vernal Equinox direction, also known as the direction of the first point of Ares.

The axes of the ECEI coordinate frame share the naming convention of those in the ECEF frame.

1.6.4 Body Fixed In-Track, Cross-Track, Radial (LVLH)

This coordinate frame is used to describe orbital position errors. The cross-track component is in the direction of the orbital angular momentum ($\vec{h}$) direction. The radial component is in the direction of the line connecting the center of Earth and the spacecraft. The in-track component completes the triad and is mostly in the velocity vector direction for near circular orbits.

Chapter 2: Orbit Determination

As mentioned previously, orbit position knowledge is required, and the standard solutions of orbit propagation with SGP4 or GPS are not applicable. Therefore, a simple orbit determination scheme is needed; the following section presents a simplified orbit propagator for this purpose.

2.1 Orbit Propagator Design and Results

The basic concept of the orbit propagator, as mentioned previously, is a circular trajectory with position on the trajectory defined by angular velocity times time. Similar to defining an orbit by Classical Orbital Elements (COEs), several parameters are required to define the trajectory and the spacecraft position on the trajectory; however, in this simplified scenario, the parameters are a simplified subset of the COEs.

First, the orbital plane—the plane of the circular trajectory—must be defined; this is accomplished through the angular velocity vector $\vec{h}$. Since $\vec{h}$ is a vector in 3-space it is defined by 3 scalar values; however, since we are only concerned with the direction of $\vec{h}$—which is normal to the orbital plane—we are only concerned with the unit vector $\hat{h}$, which is the vector in the direction of $\vec{h}$ with magnitude 1. The variable $\hat{h}$ is a unit vector in 3-space and therefore is defined by two scalar values ($\hat{h}_{\hat{\imath}}$, $\hat{h}_{\hat{\jmath}}$).

Unlike with the COEs, the orbital shape does not need to be defined; since it is fixed as a circle, eccentricity is zero and the relevant equations can be simplified accordingly. Therefore, the second parameter to define is the orbit size. The orbit size is defined by the radius of the circle, R, which is equivalent to the semimajor axis of a circular orbit.

Finally, the spacecraft position must be defined. This is defined by two parameters: argument of latitude—similar to true anomaly but referenced from the ascending node—of the spacecraft at

the beginning of the orbit propagation window (u), and the angular velocity of the spacecraft's orbit about Earth, (ω). This allows the spacecraft's position to be simply defined by the equation

$$u(t) = \nu_{initial} + \omega * t \quad \textbf{1}$$

where $u(t)$ is the argument of latitude at time t.

This model defines the spacecraft's orbit by 5 parameters: $\hat{h}_{\hat{I}}$, $\hat{h}_{\hat{J}}$, R, $u_{initial}$,and ω. The orbit's eccentricity and all perturbations to the orbit are ignored. Equations 2-4 show the calculations required to determine ECEI position from time and the 5 parameters mentioned.

$$R_{\hat{I}} = -\frac{R * \left(\hat{h}_{\hat{J}} * \cos(u) + \hat{h}_{\hat{I}} * \hat{h}_{\hat{K}} * \sin(u)\right)}{\sqrt{\hat{h}_{\hat{I}}^{\,2} * \hat{h}_{\hat{J}}^{\,2}}} \quad \textbf{2}$$

$$R_{\hat{J}} = \frac{R * \left(\hat{h}_{\hat{I}} * \cos(u) - \hat{h}_{\hat{J}} * \hat{h}_{\hat{K}} * \sin(u)\right)}{\sqrt{\hat{h}_{\hat{I}}^{\,2} * \hat{h}_{\hat{J}}^{\,2}}} \quad \textbf{3}$$

$$R_{\hat{K}} = R * \sin(u) * \sqrt{\hat{h}_{\hat{I}}^{\,2} * \hat{h}_{\hat{J}}^{\,2}} \quad \textbf{4}$$

Equations 2-4 do contain a single unique sine, cosine and square root, and these functions are not natively supported by the processors being considered. However, it will be discussed later that relatively low computational cost solutions exist for evaluating trigonometric functions on low power fixed point processors. Equations 2-4 are not applicable to near equatorial orbits because of a singularity in the representation; however, CubeSats have not been put in equatorial orbits and if they were, it would be simple to change the representation.

An observant reader will also notice that Equations 2-4 include $\hat{h}_{\hat{K}}$; this value will either need to be computed from $\hat{h}_{\hat{I}}$ and $\hat{h}_{\hat{J}}$ or uplinked as a sixth parameter. The unit length of $\hat{h}$ allows the simple calculation of $\hat{h}_{\hat{K}}$ with the equation:

$$\hat{h}_{\hat{K}} = \sqrt{1 - \hat{h}_{\hat{I}}^{\,2} + \hat{h}_{\hat{J}}^{\,2}} \qquad 5$$

Equation 5 adds a square root—potentially a computationally expensive operation—and uplinking $\hat{h}_{\hat{K}}$ increases the total number of uplink parameters for a tight uplink budget. This trade will need to be made with the uplink characteristics and square root routine in mind. This decision may be obvious or trivial if uplink is abundant or an efficient square root algorithm is used.

Figure 1 shows the position error of this model when compared to SGP4 for the CP3 orbit between 12:00 UTC 01-23-2008 and 12:00 UTC 01-25-2009. This orbit is characteristic of most CubeSats that have been launched, and it will be used for comparison purposes throughout this report. Appendix A contains the CP3 TLE used here and more information about CubeSat orbits. When creating this figure an optimization routine was used; the optimization is discussed in Section 2.2. Figure 2 shows the same method displayed in Figure 1 without this optimization routine; clearly there is much more error.

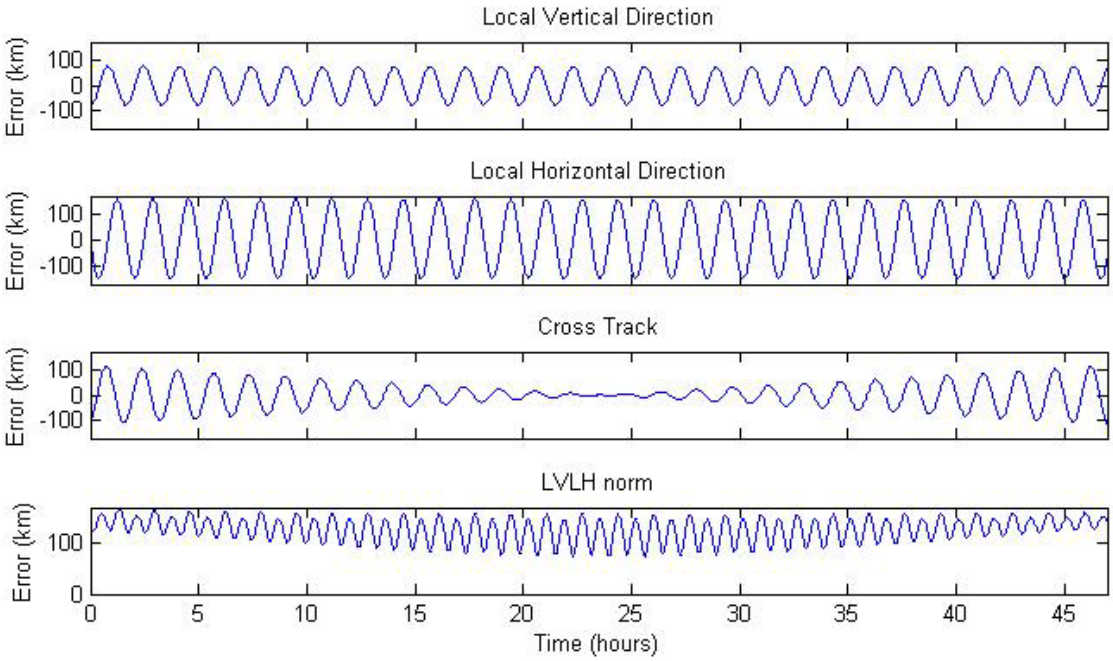

Figure 1 Orbit propagator (5 parameters)

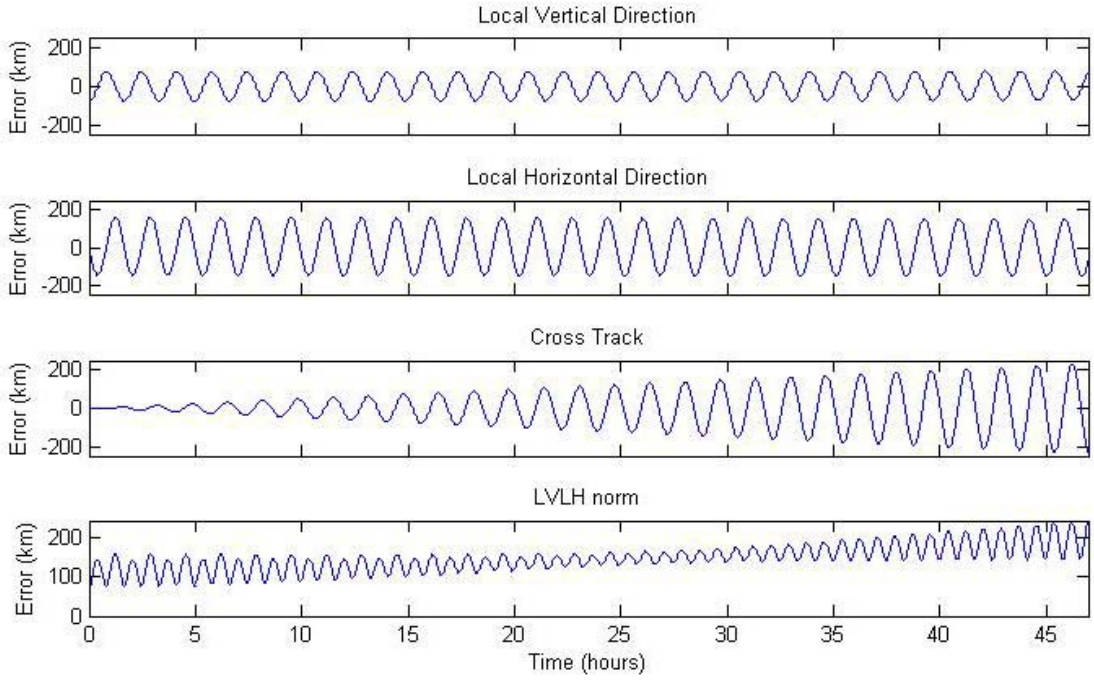

Figure 2 Orbit propagator (5 parameters) no optimization

Figure 1, and the figures to follow in this section, display error in the LVLH coordinate frame—explained in section 1.6—as well as the total error. These errors represent the difference between SGP4 and the orbit propagation model; that is, they represent the vector between the two positions and the components of this vector. Cyclic perturbations can be seen in all three directions: local vertical, local horizontal and cross-track. Secular perturbations can be seen in the cross-track error. The average difference between the simple propagator described above and SGP4 in the situation above was 125.4km and the maximum error was 165.1km. This is not very accurate; however, it is almost accurate enough to provide crude attitude knowledge from Earth referencing sensors.

The next step is to identify the major sources of error and try to minimize them in a computationally efficient way.

The dominant perturbation in the cross-track direction is theorized to be the regression of the nodes, $\dot{\Omega}$. In the 5 parameter model the orbital plane is assumed to be fixed in inertial space since $\hat{h}$ is constant. In reality, the oblateness of the Earth causes the orbital plane to precess[4], which is apparent in the regression of the nodes. A $\dot{\Omega}$ term is added to the orbit model to correct for the precession of the orbital plane. This translates into a rotation of the $\hat{h}$ vector about the $\hat{K}$ axis. The orbital model now consists of 6 parameters: $\hat{h}_I$, $\hat{h}_J$, R, $u_{initial}$, ω, and the additional term $\dot{\Omega}$. Computing the right ascension of the ascending node Ω from $\hat{h}_I$ and $\hat{h}_J$ is possible, but this requires an inverse tangent—or other inverse trigonometric function—which is a computational burden. For this reason, the 6 parameter model is described by R, $u_{initial}$, ω, $\dot{\Omega}$, Ω, and $\sqrt{\hat{h}_I^{\,2} + \hat{h}_J^{\,2}}$. This eliminates the need for an inverse tangent while still describing the orbital plane by two parameters. Figure 3 shows the error of the 6 parameter model compared to the same CP3 data in SGP4.

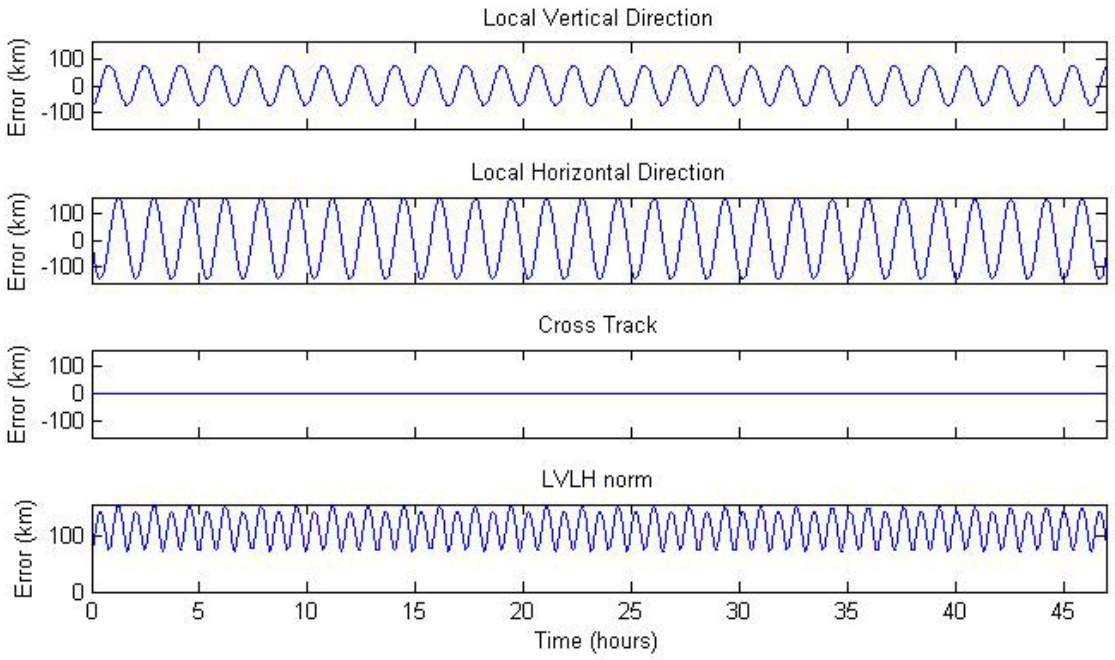

Figure 3 Orbit propagator (6 parameters)

Clearly, the error is drastically reduced in the cross-track direction, thus confirming the theory that the cross track error is predominately caused by orbital plane precession and improving the orbital model with the addition of only one parameter. The overall average and maximum errors are reduced to 115.5km and 156.1km respectively. The average cross track error is reduced from 36.3km to 0.69km, and the maximum cross track error is reduced from 111.98 to 2.18. Equations 6 and 7 below show the calculation for $\hat{h}$, which is now based on $\dot{\Omega}$ and time.

$$\Omega(t) = \Omega_{initial} + \dot{\Omega} * t \quad \textbf{6}$$

$$\hat{h} = [\hat{h}_{\hat{I}}, \hat{h}_{\hat{J}}, \hat{h}_{\hat{K}}] = \left[\cos\left(\Omega - \frac{\pi}{2}\right) * \sqrt{\hat{h}_{\hat{I}}{}^2 + \hat{h}_{\hat{J}}{}^2}, \sin\left(\Omega - \frac{\pi}{2}\right) * \sqrt{\hat{h}_{\hat{I}}{}^2 + \hat{h}_{\hat{J}}{}^2}, \hat{h}_{\hat{K}}\right] \quad \textbf{7}$$

Equation 7 does add a sine, cosine, and square root to the required calculations, and these should be kept to a minimum; however, a routine for sine, cosine and square root will already be in place, because they are needed for the 5 parameter model.

The next errors to reduce are in the local horizontal and local vertical directions. It is theorized that these periodic errors are mainly caused by the eccentricity of the orbit. The eccentricity causes a change in radius, which is apparent in the local vertical error. Eccentricity also causes a change in orbital velocity—or angular velocity (ω)—which causes an error in the local horizontal direction. These periodic errors appear mostly sinusoidal, and the proposed correction is the use of a sine function to cancel them out. A sine function is added to the angular velocity term (ω), and the radius term (R). Three parameters are required to describe a sine function: starting angle, magnitude, and frequency. In this case, the starting angle and frequency are related for the ω-sine and the R-sine because these terms are dependent on the period of the orbit and spacecraft position (u). Therefore, the addition of these two sine correction factors consists of four parameters: The frequency of the sine functions (f), the initial angle ($\theta_{initial}$), the amplitude of

the angular velocity sine (A_ω), and the magnitude of the radius sine (A_R). These parameters are factored into Equations 8 and 9.

$$u(t) = u_{initial} + \omega * t + A_\omega * \sin\left(\theta_{initial} + \frac{\pi}{2} + f * t\right) \quad \textbf{8}$$

$$R(t) = R + A_R * \sin\left(\theta_{initial} - \frac{\pi}{2} + f * t\right) \quad \textbf{9}$$

Equations 8 and 9 do add a sine evaluation; however, they reduce the average error and maximum error to 1.47km and 2.67km respectively. This result is shown in Figure 4, note the change in y-axis scale.

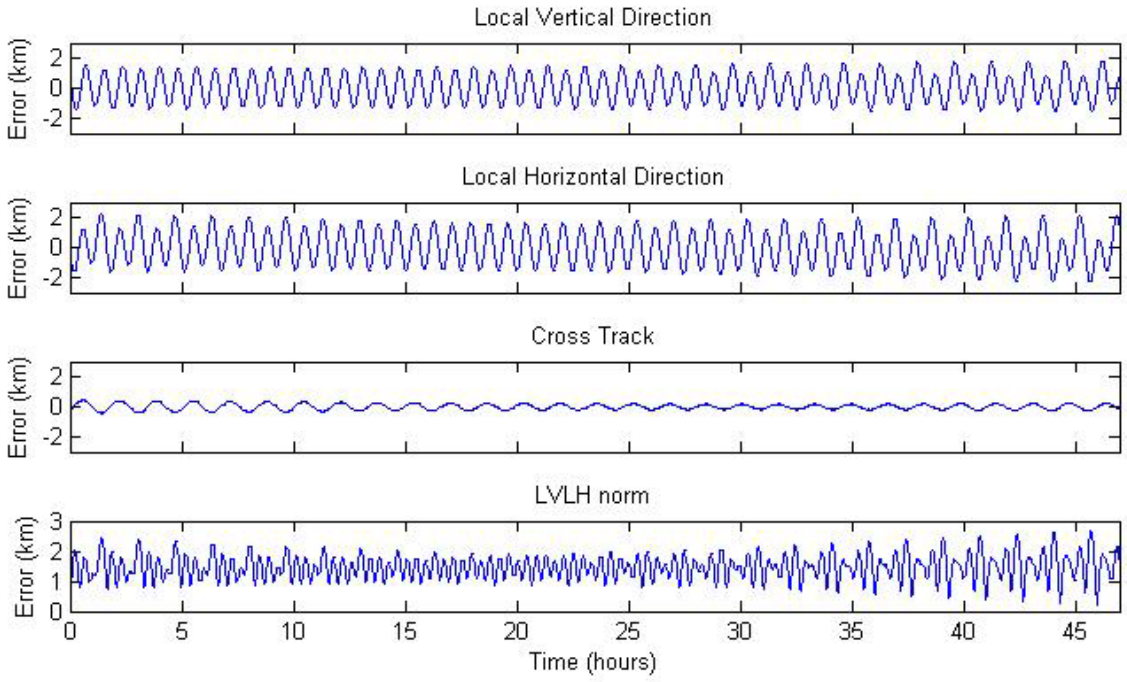

Figure 4 Orbit propagator (10 parameters)

This orbital model consists of 10 parameters: r, $v_{initial}$, ω, $\dot{\Omega}$, Ω, $\sqrt{\hat{h}_{\hat{\imath}}{}^2 + \hat{h}_{\hat{\jmath}}{}^2}$, f, $\theta_{initial}$, A_ω, and A_R. Table 2 shows the error associated with each of these methods when propagated for 2 days, it also shows that the 10 parameter model is stable, since the error did not drastically increase on the

14 day case. Other satellites—including Compass One, GeneSat-1, and XI-V—yielded similar results; therefore, CP3 will be used as the test case for the majority of the work to follow.

Table 2 Orbit propagation results

Satellite	Duration (days)	Propagator	Average Error (km)	Maximum Error (km)
CP3	2	5 parameters	125.4	165.1
CP3	2	6 parameters	115.5	156.1
CP3	2	10 parameters	1.471	2.673
CP3	14	10 parameters	3.700	10.73

2.2 Optimization of Orbit Propagator Parameters

The orbit propagator and the results shown previously are only possible with the careful optimization of the orbit propagation parameters. The model described in the previous section is very sensitive to small changes in the parameters. This sensitivity makes finding the optimum parameters very difficult. The model requires groups of 5, 6 or 10 parameters—which suggests multivariable optimization. However, this is difficult with all the parameters being so independently sensitive. Initial attempts at optimization yielded divergence or inconsistent results.

Optimization was performed with the MATLAB functions “fminunc” and “fmincon”. These two functions attempt to find the minimum of multivariable scalar functions starting at an initial condition. “fminunc” is an unconstrained optimizer, while “fmincon” is a constrained optimizer. In this case, the function to be minimized—or cost function—is the orbit propagation model with the inputs of the orbit propagator parameters, and truth data from SGP4 and the output is error. The error used as the “cost” is the mean of the Euclidian norm of the difference between the propagator position vector and the SGP4 position vector.

$$cost = \sqrt{\left(R_{\hat{I}_{prop}} - R_{\hat{I}_{SGP4}}\right)^2 + \left(R_{\hat{J}_{prop}} - R_{\hat{J}_{SGP4}}\right)^2 + \left(R_{\hat{K}_{prop}} - R_{\hat{K}_{SGP4}}\right)^2} \quad \textbf{10}$$

This error is not the true position error—which cannot be known for this application—but the error with respect to an accurate enough reference frame. SGP4 is considered accurate enough because errors associated with it—for the time frames being considered—are at least one order of magnitude less than the target error. Consider, for example, the error associated with 1 second of clock error is several kilometers, which is similar to the accuracy of SGP4 over a period of days.

To ensure convergence, the optimization is done in a stepwise manner with all of the variables being initialized to the best available accuracy. MATLAB's "fminunc" is used when convergence is consistent, "fmincon" when it is not. MATLAB's "fmincon" aids in ensuring convergence by bounding the results with respect to the initial condition. However, if the constraints are active the solution may not be optimal; therefore, "fmincon" is only used when required.

The base orbit propagator is described by 5 parameters, and these must be optimized together, since error cannot be determined without all of them. This is the source of the convergence difficulties; optimizing 5 parameters is possible—and entirely within the scope of MATLAB's optimization routines—but convergence is poor in this particular case.

The radius parameter is found from the average radius output by SGP4 for the propagation window of interest. The initial $\hat{h}$ vector is found by crossing the first radius vector and the radius vector one quarter of an orbit later. The angular velocity (ω) is found by averaging the angular change between SGP4 radius vectors. The argument of latitude is initialized to zero and this is quickly remedied with optimization in the next step.

Once the 5 parameters are initialized, the stepwise optimization begins with the argument of latitude followed by angular velocity—optimized with the new value for true anomaly—and then

the h vector optimized with the new value for argument of latitude and angular velocity. At each of these steps the orbital model is run through the optimizer with the initialized or optimized parameters and average error over the orbit is minimized. Finally, with the optimized parameters, the entire model can be optimized with all 5 variables. Convergence is ensured by the good initial condition provided by the prior optimization and the use of bounds in "fmincon". This yields an optimized 5 parameter model. In this scenario, the h vector changes to describe the orbital plane in the middle of the propagation window; this minimizes the cross track error for this particular scenario. This is shown in Figure 1 where the cross track error first decreases, reaches a minimum in the middle of the propagation window and then grows to a level similar to the starting error.

For the 6 parameter model $\dot{\Omega}$ needs to be initialized. The orbital model is particularly sensitive to small changes in $\dot{\Omega}$. Prussing[4] explains that the regression of the nodes is mainly caused by the asphericity of Earth; particularly Earth's oblateness—described by the J_2 gravitational coefficient—since J_2 is of order 1000 times larger than any other gravitational coefficients. The average rate of nodal regression can be approximated by the J_2 coefficient, the semilatus rectum (p), the mean motion (n), the radius of Earth (R_{Earth}), and the inclination (i), this is shown in Equation 11.

$$\dot{\Omega}_{AV} = -\frac{3n}{2} J_2 \left(\frac{R_{Earth}}{p}\right)^2 \cos(i) \qquad \textbf{11}$$

In optimizing the 5 parameter case, $\hat{h}$ was changed from the initial $\hat{h}$ vector to the $\hat{h}$ vector in the middle of the propagation window. Changing the $\hat{h}$ vector minimized the error for that particular case; however, with the addition of the $\dot{\Omega}$ term, $\hat{h}$ must describe the orbital plane at the beginning of the propagation window. This minimizes the initial cross track error and $\dot{\Omega}$ maintains the orbital plane in the correct location. To accomplish this, the optimization of $\hat{h}$ is postponed until after $\dot{\Omega}$ has been calculated and optimized. Once all of the parameters are initialized through

individual optimization, they can be optimized together—with the assistance of bounds. This yields the optimized 6 parameter method.

The 10 parameter model includes the addition of the parameters f, $\theta_{initial}$, A_ω, and A_R. The propagation window is started at perigee; this makes initializing $\theta_{initial}$ much more straightforward. The parameter $\theta_{initial}$ defines the starting point of both the angular velocity sine and the radius sine. These two parameters are related, but offset from each other by a factor of π. At perigee, radius is minimized and the angular velocity is maximized—the converse is true at apogee. Fixed offsets are added to $\theta_{initial}$ to achieve the desired starting values for ω and R. This produces equivalent results to individually optimizing two angle parameters, and reduces the uplink requirement. The frequency term (f) is defined as $f = \frac{2\pi}{T}$ where T is the period. The amplitude of the sine functions are defined by half the difference between the maximum and minimum values from SGP4,

$$A_R = \frac{R_{max} - R_{min}}{2} \quad \textbf{12}$$

$$A_\omega = \frac{\omega_{max} - \omega_{min}}{2} \quad \textbf{13}$$

for R and ω respectively. The model is very sensitive to small changes in frequency (f), similar to the sensitivity to $\dot{\Omega}$; this is because a small offset in either of these parameters will grow into a large offset over time. Errors in these parameters cause a secular error, whereas errors in other terms—like radius magnitude for example—will cause a bounded periodic error. Frequency must be optimized by itself because of this sensitivity. Once the frequency is optimized, all 10 parameters can be optimized together yielding parameters for the final model presented in 2.1.

This optimization scheme consistently converges and produces good results for typical CubeSat orbits; however, convergence is not guaranteed. In some situations the maximum number of

function evaluations may need to be increased to achieve the best solution. Though not typical, the bounds may need to be modified if constraints are active at optimization termination. These concerns make this a good working solution but not a "solve all," since user intervention and knowledge may be required. Also, other orbits may require a slightly different approach to initialization or bounding. To maintain orbit determination knowledge, this routine needs to be run frequently and the new parameters need to be uplinked to the spacecraft.

Chapter 3: Vector Observations

3.1 Geomagnetic field

3.1.1 Magnetometer Background

Magnetometers provide small, light, and inexpensive vector observation for LEO spacecraft—useful below 6000 km altitude[1]—which has led to their use in many spacecraft. Magnetometers typically[1] provide accuracy between 5° and 1°; therefore, they are used on spacecraft that only demand attitude knowledge accurate to degrees, or they are used as a secondary reading on spacecraft with more demanding requirements. Spacecraft which require sub-degree pointing accuracy still require crude attitude knowledge for initial attitude determination, lost in space scenarios, and as a subset of more complex routines. The CPX bus includes either 5 or 6—depending on the configuration—2-axis magnetometers which yield 3 or 4 readings in each of the 3 body axes. These sensors are a small surface mount package which allows them to be placed on the spacecraft's exterior panels; they have gained flight heritage in this configuration on CP3, CP4 and CP6. Many other magnetometers—including 3-axis sensors and more accurate sensors—are currently being produced; the combination of their low cost, accuracy, and small package makes them ideal for CubeSat applications. Also, magnetorquers are a common attitude control method, and they generally require magnetometers; therefore, a spacecraft with magnetorquers will already have magnetometers available for attitude determination.

Magnetometers—like all attitude sensors—do have some drawbacks. The Earth's magnetic field has been thoroughly studied and is modeled accurately; however, it is variable and cannot be modeled perfectly. Also, the electronics of a spacecraft produce a magnetic field which will corrupt magnetometer data. Electronics can be designed with an effort to minimize their effect on magnetometer readings, and their effect can be characterized; however, they will inevitably

induce some error, especially in CubeSats where the interference is poorly controlled and modeled. Also, the magnetic field needs to be known for all possible positions in the orbit; therefore, the spacecraft must have orbit determination and a magnetic field model.

3.1.2 IGRF 10

The most widely used magnetic field model is the International Geomagnetic Reference Field, of which the tenth generation—IGRF 10—is valid for 1900.0 to 2010.0. This standard mathematical description of the Earth's main magnetic field is compiled and published by the International Association of Geomagnetism and Aeronomy[5] (IAGA).

The IGRF 10 model was used in this work as the most accurate source of Earth's magnetic field information. This data will be used to develop less computationally demanding models, and to compare accuracy. The following is a summary of the IGRF model as relevant to this work.

In the IGRF models, the IAGA describes the Earth's main field by series of coefficients—195 coefficients and 80 terms describing secular variation for IGRF 10—which are the coefficients of a truncated harmonic series. In this method, the Earth's magnetic field, B, is expressed as the gradient of a scalar potential function V.

$$B = -\nabla V \tag{14}$$

V is represented by a truncated series of spherical harmonics as[6]:

$$V(r,\theta,\phi) = a\sum_{n=1}^{k}\left(\frac{a}{r}\right)^{n+1}\sum_{m=0}^{n}\left(g_n^m\cos(m\phi)+h_n^m\sin(m\phi)\right)P_n^m(\theta) \tag{15}$$

Where a is the magnetic reference radius—6371.2 km— adopted by IAGA; g_n^m and h_n^m are Gaussian coefficients; $P_n^m(\theta)$ are the Schmidt normalized associated Legendre functions; and r,

θ, and ϕ are the geocentric coordinates: radius, coelevation, and longitude. In this notation the subscript "n" denotes the degree and the superscript "m" denotes the order. Note, the IAGA documentation uses slightly different notation.

The IAGA documentation leaves the IGRF model there; however, there is a significant amount of math that has to be done to evaluate Equation 15. There are multiple implementations of the IGRF model available in multiple programming languages; however, it was in the best interest of this work to develop a specialized routine, both because the available routines are typically "black box" with little control or transparency, and because computational efficiency is important. Although simpler solutions will be developed in the following sections, it is the goal of this section to develop code for the IGRF model that could be implemented on a CubeSat if the accuracy is deemed worth the computational burden. With computational efficiency in mind, the recursive relations presented in Wertz[6] were selected.

To develop the recursive relations—and in turn reduce the computational burden—Wertz[6] converts the Schmidt normalized Legendre functions $P_m^m(\theta)$ to the Gauss normalized functions $P^{n,m}(\theta)$ through Equation 16.

$$P_n^m = S_{n,m} P^{n,m} \tag{16}$$

Wertz suggests combining the $S_{n,m}$ factors with the Gaussian coefficients because neither of them are dependent on position. This means they only need to be calculated once; however, the Gaussian coefficients are a function of time through the secular variation terms. The Gaussian coefficients—and therefore the combined $S_{n,m}$ factors and Gaussian coefficients—would need to be recalculated each time step if this model were to be implemented at full accuracy. However, the Gaussian coefficients change very slowly; a slightly lower fidelity and simpler model could assume constant Gaussian coefficients and calculate the combined $S_{n,m}$ factors and Gaussian coefficients on the ground, or at smaller intervals. Wertz combines these terms as follows:

$$g^{n,m} \equiv S_{n,m} g_n^m \quad \textbf{17}$$

$$h^{n,m} \equiv S_{n,m} h_n^m \quad \textbf{18}$$

and presents the following recursive formulae for $S_{n,m}$

$$S_{0,0} = 1 \quad \textbf{19}$$

$$S_{n,0} = S_{n-1,0}\left(\frac{2n-1}{n}\right) \qquad n \geq 1 \quad \textbf{20}$$

$$S_{n,m} = S_{n,m-1}\sqrt{\frac{(n-m+1)(\delta_m^1+1)}{n+m}} \qquad m \geq 1 \quad \textbf{21}$$

where δ_m^1 represents the Kronecker delta.

Wertz also provides recursive relations for the $P^{n,m}$ term and the partial derivatives of $P^{n,m}$, $\frac{\partial P^{n,m}}{\partial \theta}$, which are needed because of the gradient in Equation 14.

$$P^{0,0} = 1 \quad \textbf{22}$$

$$P^{n,n} = \sin(\theta)\, P^{n-1,n-1} \quad \textbf{23}$$

$$P^{n,m} = \cos(\theta)\, P^{n-1,m} - K^{n,m} P^{n-2,m} \quad \textbf{24}$$

$$\frac{\partial P^{0,0}}{\partial \theta} = 0 \quad \textbf{25}$$

$$\frac{\partial P^{n,n}}{\partial \theta} = \sin(\theta)\frac{\partial P^{n-1,n-1}}{\partial \theta} + \cos(\theta)\, P^{n-1,n-1} \qquad n \geq 1 \quad \textbf{26}$$

$$\frac{\partial P^{n,m}}{\partial \theta} = \cos(\theta)\frac{\partial P^{n-1,m}}{\partial \theta} - \sin(\theta)\, P^{n-1,m} - K^{n,m}\frac{\partial P^{n-2,m}}{\partial \theta} \quad \textbf{27}$$

where

$$K^{n,m} \equiv \frac{(n-1)^2 - m^2}{(2n-1)(2n-3)} \qquad n > 1 \tag{28}$$

$$K^{n,m} \equiv 0 \qquad n = 1 \tag{29}$$

Finally, with the results from above, the magnetic field can be calculated for a given position with Equations 30-32.

$$B_r = \frac{-\partial V}{\partial r} = \sum_{n=1}^{k} \left(\frac{a}{r}\right)^{n+2} (n+1) \sum_{m=0}^{n} \left(g^{n,m} \cos(m\phi) + h^{n,m} \sin(m\phi)\right) P^{n,m}(\theta) \tag{30}$$

$$B_\theta = \frac{-1}{r} \frac{\partial V}{\partial \theta} = -\sum_{n=1}^{k} \left(\frac{a}{r}\right)^{n+2} \sum_{m=0}^{n} \left(g^{n,m} \cos(m\phi) + h^{n,m} \sin(m\phi)\right) \frac{\partial P^{n,m}(\theta)}{\partial \theta} \tag{31}$$

$$B_\phi = \frac{-1}{r \sin(\theta)} \frac{\partial V}{\partial \phi} = \frac{-1}{\sin(\theta)} \sum_{n=1}^{k} \left(\frac{a}{r}\right)^{n+2} \sum_{m=0}^{n} m\left(-g^{n,m} \sin(m\phi) + h^{n,m} \cos(m\phi)\right) P^{n,m}(\theta) \tag{32}$$

To reduce the computational requirement, Wertz suggests using recursive relations for $\sin(m\phi)$ and $\cos(m\phi)$ terms in Equations 30-32. The recursive relations are:

$$\cos(m\phi) = \cos\big((m-1)\phi\big)\cos(\phi) - \sin\big((m-1)\phi\big)\sin(\phi) \tag{33}$$

$$\sin(m\phi) = \sin\big((m-1)\phi\big)\cos(\phi) + \cos\big((m-1)\phi\big)\sin(\phi) \tag{34}$$

which allow all of the $\sin(m\phi)$ and $\cos(m\phi)$ terms in Equations 30-32 to be evaluated with only two trigonometric function evaluations.

These results are in the LLR coordinate frame, and must be transformed into ECEF and then to ECEI when using inertial coordinates for analysis. The above process for determining the Earth's

magnetic field by use of spherical harmonics and the IGRF coefficients has been simplified through the recursive relations presented in Wertz; however, this is still relatively complex. The model—as coded from the above method—requires multiple sets of nested loops to calculate the Gaussian coefficients, Legendre functions and their derivatives, and the final magnetic field values. This process can be simplified—for a loss of accuracy—through reducing the degree and order of the model, or by assuming the Gaussian coefficients are constant, but ultimately the simplification would be minor.

This model—as coded in MATLAB—was checked with online calculators available through NOAA and other sources and proved to yield the expected results. One unexpected aspect of this model—and the other models publicly available—is a singularity over the poles. Equation 32 returns an infinite—and meaningless—value when the sine of the coelevation goes to zero, namely at the poles. There are a variety of ways to get around this singularity; however, this should be considered in future implementations of this routine and in its use later on in this thesis.

Sources vary on the accuracy of the IGRF model: the IAGA publication quotes the coefficients to an accuracy of 0.1nT for the current model but does not suggest a total error; Lowes[7] suggests a root mean squared error (RMS) of 20nT for applications that are extrapolating forward like this one but does not provide a covariance. This yields a worst case scenario of approximately 0.034 degree average error for a Sun-synchronous orbit. This particular calculation is worst case because it uses the lowest magnitude observation for a test LEO Sun-synchronous orbit and because the RMS error is quoted for the surface of Earth which has a stronger magnetic field. Error in the IGRF model will be dependent on the orbit, and other less known factors like atmospheric conditions and future secular variations; however, a 0.034 degree average error will serve to quantify the error expected from this model.

3.1.3 Look-Up tables

Since the IGRF model is computationally demanding, other methods were explored in an effort to reduce the complexity of calculating the geomagnetic field on-board. One way to reduce the complexity on orbit is to make a look-up table. This reduces complexity on orbit—at the expense of data space memory—and can provide efficient solutions; however, an efficient look-up table scheme is needed. For a look up table to be efficient, it must fit within the memory allocated to it and must be indexed in a predictable fashion. If a look-up table can be directly indexed into—for example by knowing that the indices are integer multiples of a known constant—then it is extremely efficient, so efficient that it is approximately equivalent to a single arithmetic operation. Clearly, if a lookup table can provide the accuracy required and be easily indexed, it would provide a solution that is faster than any analytic solution.

For this particular work, the orbit was simplified to a sphere of radius equal to the semi-major axis; the main dilemma was distributing points on this sphere. It is not a trivial exercise to evenly distribute points on a sphere while maintaining a simple indexing scheme.

3.1.3.1 LLR Look-Up Table

The first geomagnetic field look-up table implemented relied on spherical coordinates, the LLR coordinate frame. This is convenient because the IGRF model takes the input of LLR position and outputs the magnetic field in the LLR directions. This also allows for the look-up table to simply be a function of two variables: coelevation and longitude. Points were distributed at even increments of latitude and longitude. This yields an uneven distribution on the surface of the sphere. However, this method is simple and straightforward to implement. In addition, the steepest field gradient—which occurs over the poles—coincides with the high table density at the poles. Finally, the highest density points about the poles are not required for a typical CubeSat mission, because the highest inclination to date is ~98°; thus eliminating the need for the points with coelevation outside the 8° to 172° range.

This table was created by segmenting the 0°-180° coelevations into $n + 1$ values, and the 0°-360° longitudes into $2n$ points. Next, the look-up table was created by evaluating the IGRF model at the $2n(n + 1)$ points, which correspond to the latitude and longitude values. The geomagnetic field is relatively smooth and therefore lends itself to linear interpolation. Any location in coelevation and longitude can be translated into an index in the look-up table; the ceiling and floor of this index can then be found and the four combinations of coelevation and longitude ceilings and floors provide the indices of the four neighboring look-up table values. These values are then linearly interpolated to yield the magnetic field at the desired location. The interpolation scheme—while requiring several operations—drastically reduces the number of data points needed for a given accuracy when compared to slightly simpler routines, such as choosing the nearest neighbor. For example, simply rounding to the nearest point with $n = 180$—convenient 1° degree increments—provides an average and maximum angular error of 0.264° and 1.30° respectively; interpolating with $n = 15$—less than one hundredth the points—provides average and maximum angular errors of 0.21° and 1.59° respectively, comparable to the results of the rounding scheme with $n = 180$.

The following table shows a summary of the errors for a variety of look-up table sizes. This data is taken from a comparison of the interpolation method described above and the IGRF model over the two day CP3 data set described in Appendix A.

Table 3 LLR magnetic field look-up table results

n	Total Number of Data Points	Angular Error [degrees]	
		Mean	Maximum
1000	2002000	0.088	0.571
360	259920	0.088	0.570
180	65160	0.088	0.570
100	20200	0.088	0.557
45	4140	0.088	0.674
30	1860	0.093	0.888
20	840	0.125	1.393
15	480	0.205	2.001
10	220	0.470	3.711
5	60	1.794	12.989

As expected, Table 3 shows that angular error decreases as the number of data points in the model increases. The angular error asymptotically approaches a minimum as n approaches 100; the minimum error that this model approaches is not zero because the radius of the values in the look-up table typically does not equal that of the spacecraft location. To verify that the only source of error was the difference in radius, the "truth" from IGRF was calculated at the semi-major axis and the error approached zero.

Figure 5, on the following page, shows the angular error and LLR position over one day of the CP3 data set described in Appendix A, with $n = 180$. This figure displays the distribution of error minimums and maximums; most interestingly, the increased error over the equator. This is expected because of the lower resolution of the look-up table over the equator and is clearly the case for this data set.

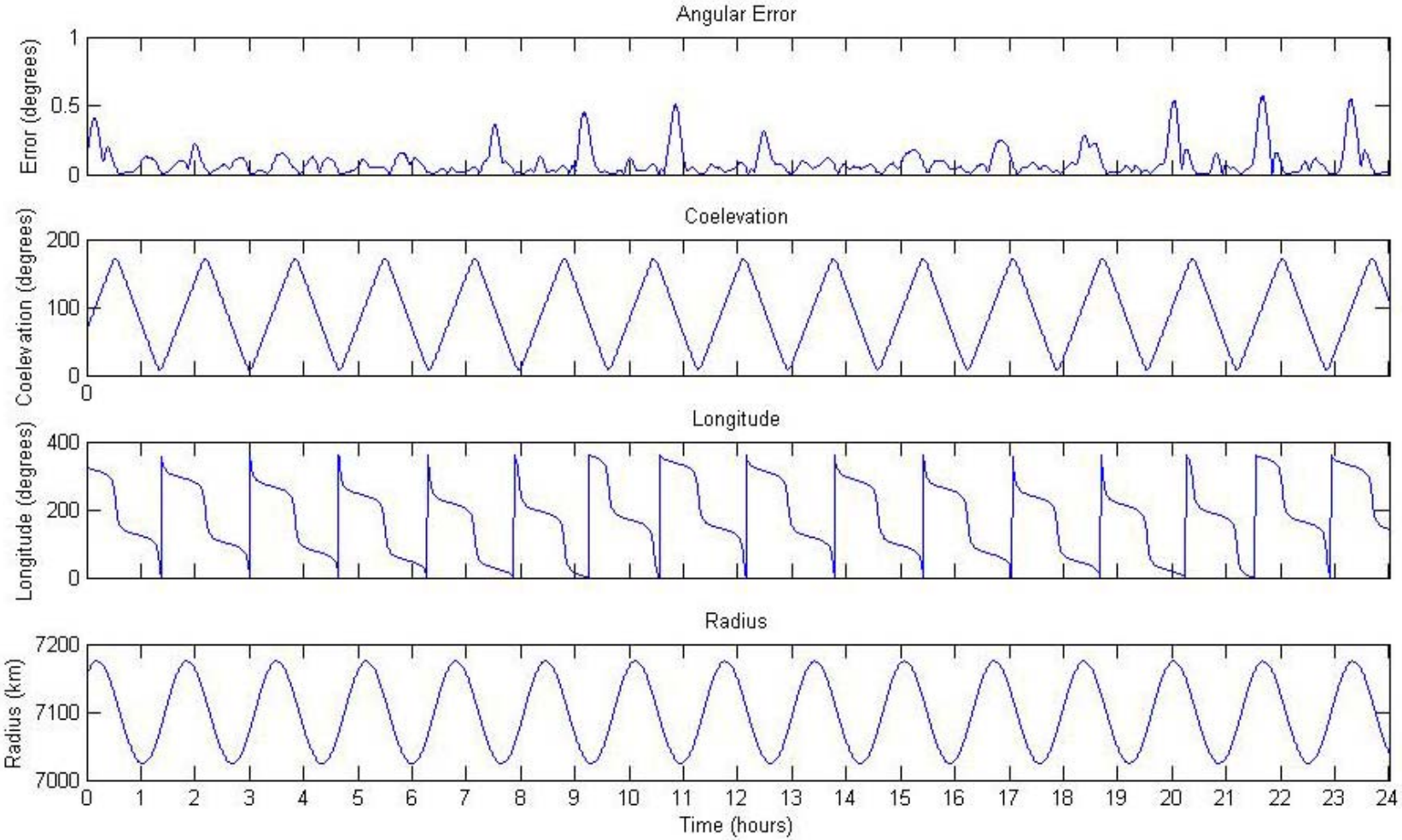

Figure 5 LLR magnetic field look-up table simulation: CP3, n=180

This model shows that a relatively low number of points—less that 1860—yields a maximum error of less than 1° and an average error of less than 0.1°. If the full accuracy of the IGRF model is not required, this model could substitute given the required data space memory.

This method does have some shortcomings. First, as mentioned previously, the distribution of points is not nearly uniform. This leads to an inefficient use of data space; however, the relatively low number of points required could prove the distribution issue moot. Second, and more importantly, this method—and the IGRF model—require the input of LLR position. To transform ECEF position to LLR position requires inverse trig functions—functions that are not natively available on most low power processors and to this point are not required in any other routines. This would mean the addition of another look-up table or polynomial expansion for inverse trigonometric functions, potentially a burden.

3.1.3.2 ECEF Look-Up Table

One of the major downfalls of both the IGRF analytic model and the LLR look-up table is the input and output in LLR coordinates. The extra coordinate transformation into and out of LLR is a computational burden, and necessitates inverse trig functions which are not needed elsewhere. A look up table scheme was developed to eliminate the LLR coordinate transformations. The output of the lookup table can easily be set to ECEF coordinates, since the calculations can be done on the ground and stored in the look-up table as ECEF values. However; the input to the look-up table is not quite as straightforward. The LLR coordinate frame provided a convenient simplification of assuming constant radius, thus reducing the look-up table to two dimensions. ECEF—and the Cartesian coordinate frame—does not provide such a straightforward simplification. The complexity and memory space of a three-dimensional look-up table are prohibitive; also, the two-dimensional approach of the previous section introduces minimal error.

The radius is held constant again to reduce the dimensionality. In this case, radius is not one of the dimensions, so all three values—ECEF X, Y, and Z—will be variable. However, the three dimensions are no longer independent. Any one direction can be expressed as a function of the other two and the radius. In this work the Z parameter was removed.

$$Z_{ECEF} = \pm\sqrt{R^2 - X_{ECEF}^2 - Y_{ECEF}^2} \quad \textbf{35}$$

In this way, the ECEF position on a sphere of constant radius R can be defined by the two variables X_{ECEF} and Y_{ECEF}. However, the sign of Z_{ECEF} is ambiguous. To deal with the sign ambiguity, the look-up table is divided into a northern hemisphere and a southern hemisphere. This requires an extra check before entering the look-up table; however, the Z_{ECEF} value is known and its sign dictates which table to use.

The points in the lookup table were distributed at even increments of X_{ECEF} and Y_{ECEF} for simplicity. This yields an uneven distribution of points on the surface similar to the uneven nature of the distribution in the LLR look-up table, but the distribution is different. In this case, the equator is very sparsely populated instead of having a tight concentration at the poles. While these sound like two different ways to describe the same distribution, they work out to be quite different. One other interesting aspect of this model is the boundary condition; with even spacing in X_{ECEF} and Y_{ECEF}, the boundary—the equator—cannot be clearly defined. For this implementation, a simple solution was used: all points with radius larger than the radius of the sphere are considered to be at the equator. For a set of X_{ECEF} and Y_{ECEF} coordinates with a magnitude greater than R, the IGRF model is called with the longitude corresponding to the X_{ECEF} and Y_{ECEF} coordinates and a coelevation of 90°. This allows for linear interpolation—although not perfect—that uses the same routine as the LLR look-up table. This is not perfect, since the value for a location on the equator may be interpolated between 3 equatorial points and one non-equatorial point, which is clearly not accurate. However, in a scenario like the one

described previously, the value is only slightly corrupted by the non-equatorial look-up table entry. Also, there is a much larger underlying problem with this model; the geometry is such that small errors in position lead to large errors at the equator. For example, consider a lookup table with radius set to 7095km and the vector $[X_{ECEF}, Y_{ECEF}]$ has magnitude 7095km—this value should lie on the equator. However, if this value is corrupted by error to have a value of 7085, significant errors will result. This will result in a coelevation error of 3°, which translates into an error in the expected Z_{ECEF} of approximately 370km. A simple survey of the magnetic field yields an average difference of over 6° in the magnetic field vector between the equator and the longitude band 3° north of it. In summary, a 10km position error—well within the error introduced by assuming a sphere—results in a 6° error; this penalty is clearly too drastic.

The following table shows the errors associated with different resolutions of this model. The Total "Useful" Data Points column represents the approximate number of points in the square look-up table that are used. A square table was used for simplicity, but a different model should be used for efficient memory use. The "useful" points are assumed to be $\frac{\pi}{4}$ of the total points since $\frac{\pi}{4}$ represents the ratio of the area of a circle to that of its circumscribing square; for Table 4, the "useful" data points value is rounded up. The empty cells in the mean error column are left blank because the polar singularity corrupts data at low resolutions.

Table 4 X-Y magnetic field look-up table results

n	Total Number of Data Points	Total "Useful" Data Points	Angular Error [degrees]	
			Mean	Maximum
1000	1000000	785399	0.102	2.794
360	129600	101788	0.136	4.934
180	32400	25447	0.187	7.252
100	10000	7854	0.275	9.783
45	2025	1591	0.510	14.436
30	900	707	0.760	18.233
20	400	315	1.110	22.064
15	225	177	1.461	25.72
10	100	79	--	29.988
5	25	20	--	38.222

Clearly, the error in this model is significantly higher, especially the maximum error. Figure 6 on the following page shows the angular error and LLR position over one day of the CP3 data set described in Appendix A, with $n = 180$. The large peaks occur consistently at the equator; the larger errors are mainly a function of this model's amplification of error at the equator.

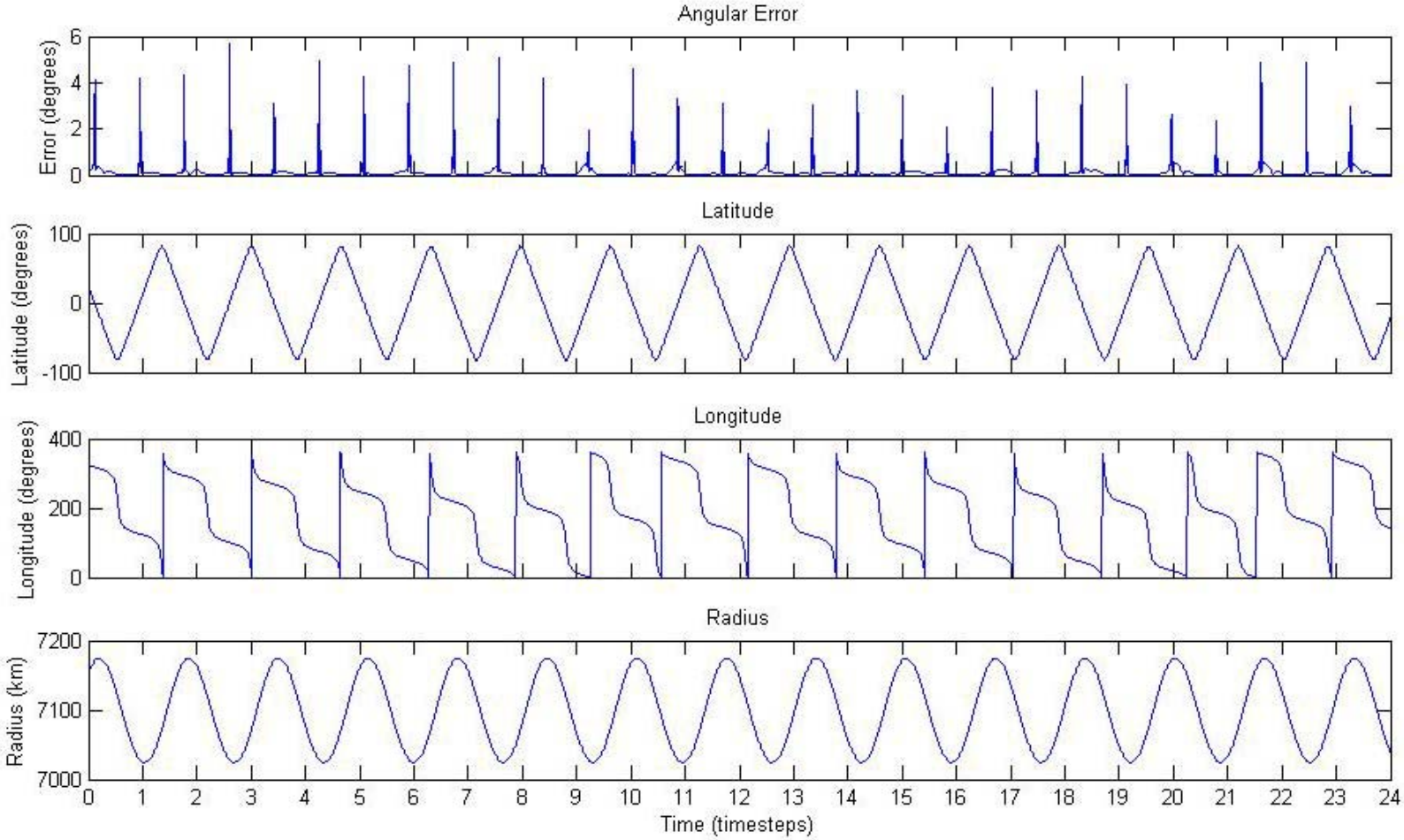

Figure 6 XY magnetic field look-up table simulation: CP3, n=180

In Figure 6 it appears that there is a singularity—or error—at the equator. However, the fact that this error approaches zero as the resolution of the model is increased suggests that the model is correct. Also, as previously mentioned, the amplification of error at the equator is expected. Figure 7 and Figure 8 show a small portion of Figure 6, approximately coelevation ranges of 70° to 110°; this shows that the error is not a single point, but a series of smooth curves. Furthermore, the "bouncing" nature of this error is characteristic of look-up tables. When a point happens to be at a look-up table value, the error is low; when a point falls at a worst case in between value the error is high. Also, not every orbit has the same error when crossing the equator, some fall in the worst case while other fall in the best, this is expected of a normally operating look-up table.

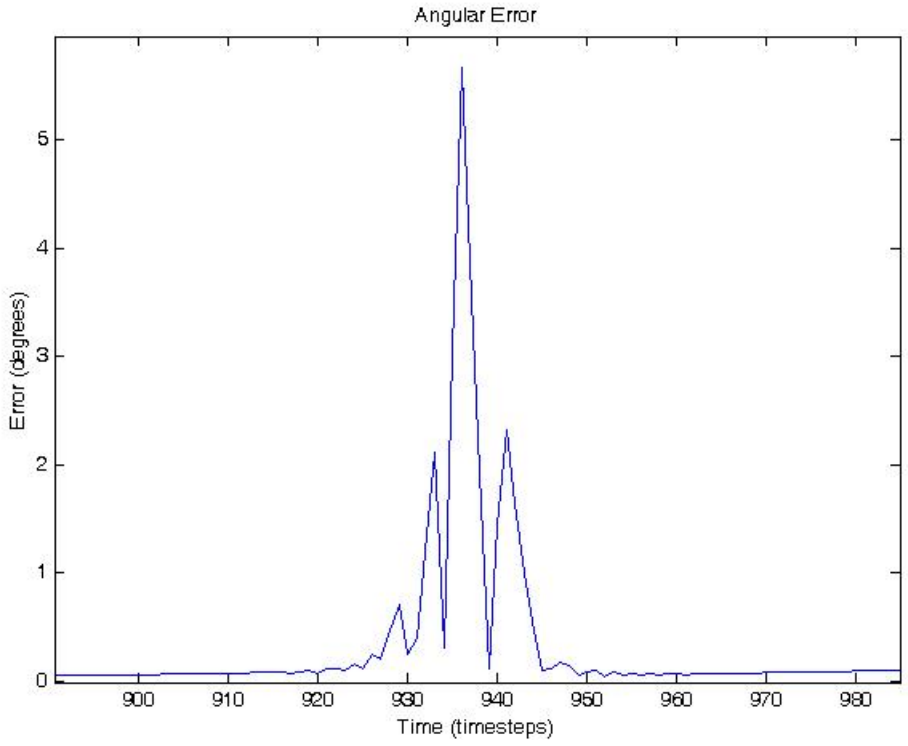

Figure 7 X-Y magnetic field look-up table simulation zoom 1

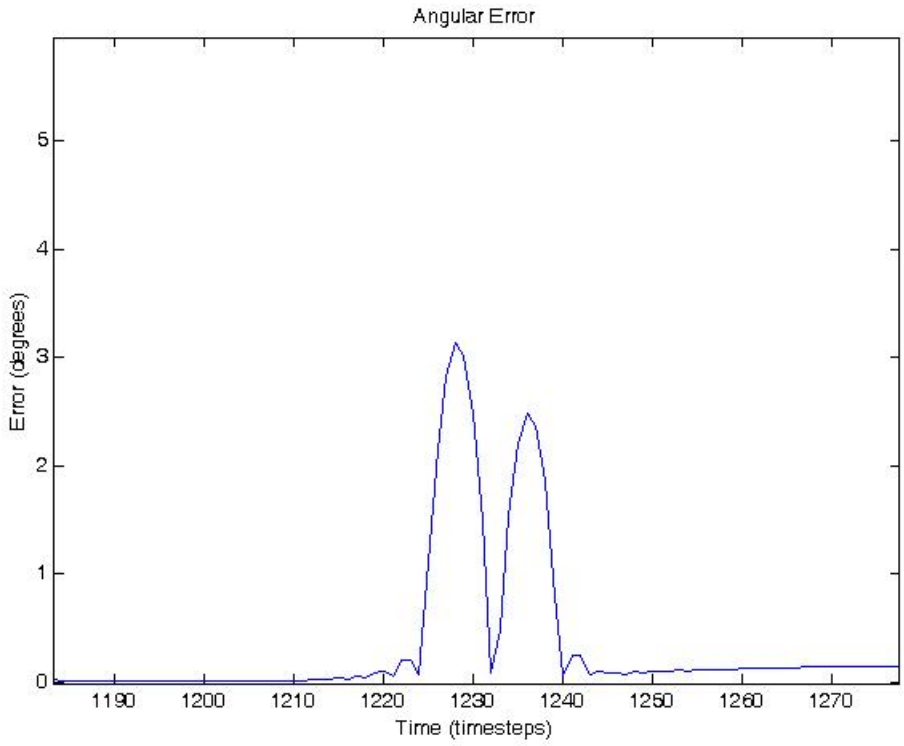

Figure 8 X-Y magnetic field look-up table simulation zoom 2

This model is not nearly as accurate as the LLR model, especially at the equator. However, if accuracy at the equator is not necessary and computational simplicity is paramount, this model could be considered. Also, the interpolation scheme and distribution of points in this model could be improved. However, this model does not lend itself to a simple nonlinear grid spacing since the boundary is ultimately not the edge of a square table, but a circle inscribed inside the square table.

3.1.3.3 Summary

While the LLR look-up table requires inverse trigonometric functions, it provided accurate solutions at minimal memory and computational cost. This model will be used in the remainder of this thesis, with the understanding that it is not perfect and could be improved or replaced. The ECEF X-Y model is presented as a template for a potential solution to the coordinate transformation problem. The IGRF model—as implemented in MATLAB from the above

approach—provides an accurate baseline to work from and its computation burden will undoubtedly eventually be overtaken by the need for better accuracy and improvement in low power processors; when this happens the presentation above will serve as a good resource.

3.2 Sun Direction

Like the magnetic field, this Sun has provided navigational guidance for hundreds of years. The simplicity and predictability that make the Sun a convenient terrestrial navigation aid also make it a convenient spacecraft attitude determination sensor. Like magnetometers, there are small and simple Sun sensor solutions available that lend themselves to CubeSat applications. However, the one major downfall of using the Sun as a reference vector is that it is not available in eclipse. However, as mentioned previously, this is not necessarily crippling, considering that attitude control—and most processes that necessitate attitude control—are power hungry, and therefore typically happen in the Sun.

The Astronomical Almanac[8] contains ephemeris data for the Sun, and also provides formulas for computing the Sun's position. The Astronomical Almanac calls these formulas "low precision" and claims a precision of 0.01°; this is a very high precision for this application and should provide more than acceptable information. The following formulas outline the method provided by the Astronomical Almanac:

The number of days since J2000.0, n, is calculated from the Julian Date:

$$n = JD - 2451545.0 \quad \textbf{36}$$

The mean longitude of the Sun, L, is given as:

$$L = 280.460° + 0.9856474°n \quad \textbf{37}$$

Similarly, the mean anomaly, g, can be calculated:

$$g = 357.528° + 0.9856003°n \quad (38)$$

L and g are used to find the ecliptic longitude:

$$\lambda = L + 1.915° \sin(g) + 0.020° \sin(2g) \quad (39)$$

The ecliptic latitude is zero:

$$\beta = 0° \quad (40)$$

The obliquity of the ecliptic, ϵ, is found from n:

$$\epsilon = 23.439° - 0.0000004°n \quad (41)$$

Finally, the direction in the Sun can be calculated in ECEI.

$$X_{ECEI} = \cos(\lambda) \quad (42)$$

$$Y_{ECEI} = \cos(\epsilon) \sin(\lambda) \quad (43)$$

$$Z_{ECEI} = \sin(\epsilon) \sin(\lambda) \quad (44)$$

The values above in red can each be removed with varying accuracy penalties, the benefit being the elimination of a trigonometric function evaluation. If trigonometric functions are computationally expensive, this is a reasonable place to trade accuracy for computational efficiency. Conveniently, all of these errors are clearly bounded.

The obliquity of the ecliptic changes slowly with time. For a 10 year period—3652.5 Julian days—the obliquity of the ecliptic changes by 0.0015°. This error could be reduced by half through "centering" the obliquity of the ecliptic in the time frame of interest. Therefore, the maximum error for a 10 year timeframe—significantly longer than most CubeSat missions—would be 0.00075°; this is well within the error allowable for this exercise. By holding this value constant, two trigonometric function evaluations are eliminated.

The terms $1.915° \sin(g) + 0.020° \sin(2g)$ in the equation for the ecliptic can also potentially be removed. The sine terms are bounded by ±1; therefore, the error associated with removing these two terms is bounded by 0.020° and 1.915° for either of the terms individually, or approximately 1.9154° maximum error if both terms are removed. This results in an average error of approximately 1.219° over an entire year.

The two models mentioned above—directly from the Astronomical Almanac and slightly simplified—will be used in the future sections; they both provide a simple solution for calculating the Sun's location, and provide another option for trading accuracy versus computational difficulty.

Chapter 4: Attitude Determination from Vector Observations

As mentioned previously, attitude determination from two vector observations provides a solution which is balanced in computational demand and hardware complexity. Many attitude determination systems require a single complicated sensor or one simple observation with complicated algorithms. Two unique vector observations can be relatively easily attained on a CubeSat, and by over-determining the problem, the computational requirement is drastically reduced.

Euler's rotation theorem tells us that a rotation—including that between a spacecraft body and an inertial frame—can be described by a single rotation about a unit vector[6]. This operation is described by three parameters: two parameters define the axis of rotation as a unit vector, and the third describes the angle by which to rotate. Two unique unit vectors—i.e. not parallel—provide four unique and relevant pieces of information and therefore over-define such a rotation; on the other hand, one vector observation only provides two pieces of information and cannot fully define orientation in 3-space without filtering over time.

Since the problem is over-defined, the general problem of finding a spacecraft's attitude from two vector observations is ultimately a matter of how to combine the available information. The first solution—TRIAD—is credited to Black[9], and it simply and elegantly combines the information; however, it somewhat arbitrarily discards information. Shortly after Black's solution was presented, Wahba[10] proposed the problem of finding a rotation matrix that minimizes a least squares cost function. This takes a step forward from TRIAD, because the extra information is discarded in such as way as to minimize error. The problem posed by Wahba has been the study of many authors, and a multitude of different solutions exist. Initial solutions found the rotation matrix as requested by Wahba. Davenport's q-method led to Shuster's QUEST algorithm[11]—

which solves for a quaternion directly. Many other methods[12], like SVD, FOAM, and ESQ, fill different niches and are applicable to specific scenarios. All of these solutions to the Wahba problem are classified as optimal solutions; they minimize the error, in turn providing the optimal solution.

Shuster[11] calls the reference unit vectors $\hat{V}_1$ and $\hat{V}_2$; for this thesis, these correspond to the inertial direction of references like the Earth's magnetic field or the Sun's direction from Earth. Shuster calls the observation unit vectors $\widehat{W}_1$ and $\widehat{W}_2$; these values will correspond to the observed vectors in the body frame—i.e. sensor readings. To define the attitude, we are looking for a rotation matrix that transforms the reference vectors to the observation vectors or vice versa. Essentially, we want to find $A = A_1 = A_2$ such that:

$$A_1\hat{V}_1 = \widehat{W}_1 \quad \textbf{45}$$

and

$$A_2\hat{V}_2 = \widehat{W}_2 \quad \textbf{46}$$

The angle between $\hat{V}_1$ and $\hat{V}_2$ should equal the angle between $\widehat{W}_1$ and $\widehat{W}_2$, since the angle between two vector observation does not depend on the reference frame—roughly speaking; however, there is an inherent error in the observed vectors because they are taken with imperfect sensors. Therefore, A_1 will not equal A_2, and a single A matrix will induce some error when used for both A_1 and A_2.

4.1 TRIAD

The TRIAD solution is presented in many references; this work will follow the notation and method presented by Shuster[11].

In the TRIAD solution, the pairs of reference and observed vectors are turned into reference and observed triads of three orthogonal unit vectors.

$$\hat{r}_1 = \hat{V}_1 \quad \textbf{47}$$

$$\hat{r}_2 = \frac{(\hat{V}_1 \times \hat{V}_2)}{|\hat{V}_1 \times \hat{V}_2|} \quad \textbf{48}$$

$$\hat{r}_3 = \hat{r}_1 \times \hat{r}_3 \quad \textbf{49}$$

$$\hat{s}_1 = \hat{W}_1 \quad \textbf{50}$$

$$\hat{s}_2 = \frac{(\hat{W}_1 \times \hat{W}_2)}{|\hat{W}_1 \times \hat{W}_2|} \quad \textbf{51}$$

$$\hat{s}_3 = \hat{s}_1 \times \hat{s}_3 \quad \textbf{52}$$

To more clearly express the A matrix as a function of the above triads, the triads are concatenated to form two matrices as follows:

$$M_{ref} = [\hat{r}_1 \vdots \hat{r}_2 \vdots \hat{r}_3] \quad \textbf{53}$$

$$M_{obs} = [\hat{s}_1 \vdots \hat{s}_2 \vdots \hat{s}_3] \quad \textbf{54}$$

where the $\hat{r}_i$ and $\hat{s}_i$ are 3 x 1 column vectors and the M matrices are 3 x 3 matrices.

Then, A can be defined as follows:

$$A = M_{obs} M_{ref}^T \quad \textbf{55}$$

The superscript "T" represents the matrix transpose operation.

The TRIAD algorithm above does not treat the "1" and "2" observations equally; in the formulation above, all of the information in the "1" vectors is used while some of the "2" vector

information is not. This means, the "1" vectors are assumed to be 100% accurate and all of the error is assumed to come from the "2" vectors. Equivalently, the formulation above produces an A matrix which satisfies Equation 45 as A_1, and finds the best A for A_2 in Equation 46 while still satisfying Equation 45. This means that the more accurate reading should be used for $\hat{V}_1$ and $\hat{W}_1$; typically, one of the sensors will be known to be more accurate and can take on the $\hat{V}_1$ and $\hat{W}_1$ values. If the values are similar in accuracy, a version of the TRIAD method called Symmetric TRIAD can be used. Symmetric TRIAD produces the equivalent solution to Wahba's problem when the variances of the two readings are equal, while TRIAD produces the same result as Wahba's problem when one reading is infinitely better than the other. However; Symmetric TRIAD is slightly more computationally demanding. Also, to justify using the Symmetric TRIAD method, the variances of the vector observations must be known, and they are typically only crudely known due to the nature of the methods and sensors employed.

It is important to note that this process does not work if the two vector observations are parallel. Mathematically, the process breaks down, because the cross product of parallel vectors results in a zero vector. As mentioned previously, 3 unique pieces of information are required to define the transformation; two parallel vectors online provide 2 values so the system is underdetermined. Depending on the types of vector observations, this may or may not be an issue. If this is an issue, it is simple to check when the vectors are aligning; readings near the singularity can be disregarded and other solutions—predicting attitude from rotation rates for example—can be used temporarily.

Also, TRIAD does not lend itself to more than 2 vector observations. While three or more vector observations may not be available initially on CubeSats, they may be eventually. The TRIAD implementations that incorporate more than two observations are cumbersome; therefore, an optimal solution should be used in these situations, since optimal solutions exist which efficiently deal with several vector observations.

The TRIAD method produces a rotation matrix—not necessarily the most efficient representation of attitude. A rotation matrix contains nine parameters; it has been shown that only 4 parameters are needed to fully define orientation in 3-space without singularity. Quaternions are the most common representation for spacecraft applications. Shuster's QUEST algorithm—and others—directly solve for the attitude quaternion, this may be desired. However, for this work, rotation matrices will suffice. If necessary, rotation matrices could be converted to quaternions and the efficiency of converting or directly solving for quaternions could be studied.

These three limitations—the non-optimality, use of rotation matrices, and the two vector limit—have led most current research and applications to optimal methods. However, there is an increased complexity and in turn computational burden with the various optimal methods. Also, the optimal methods rely on variance estimates for the observed vectors, and these values will not be accurately known on initial CubeSat implementations. For these reasons, The TRIAD method is being used in this work. The optimal methods should be reconsidered if three or more vector observations are available, the variance of the sensors is accurately known, and/or computational power is significantly increased.

Chapter 5: Orbit Determination and Attitude Determination System

Elements of a simple orbit determination and attitude determination system were presented in Chapters 2-4. Simplifications were made to meet the constraints of CubeSats, and the associated error was quantified. In this section, the entire system is combined to study how the individual errors compound. This system—the Orbit Determination and Attitude Determination System (ODAD)—is comprised of: the simplified orbit propagator, the magnetic field look-up table model, the algebraic Sun ephemeris model, and the TRIAD method. These systems—with uplinked orbit propagator parameters and accurate time—combine to provide full orbital position knowledge and 3-axis attitude knowledge. However, the accuracy of this system is less than the individual accuracies of the subsystems. Also, interactions between the subsystems must be analyzed to understand the expected accuracy and availability of the data.

5.1 Singularity

One of the largest problems with determining attitude from two vector observations is the potential for a singularity. This problem is not unique to this application—all two vector attitude determination systems are prone to singularity—but is worth analyzing as relevant to this application. In the case of a Sun-Mag system, the singularity occurs when the local magnetic field at the spacecraft location is parallel with the direction of the Sun. The geometry of the Earth's magnetic field—approximately a dipole—provides four places—assuming the field is "smooth"—where the local magnetic field is parallel to the Sun direction at any given time. Figure 9 shows a contour plot of the angle between the local magnetic field and the Sun for a LEO orbit.

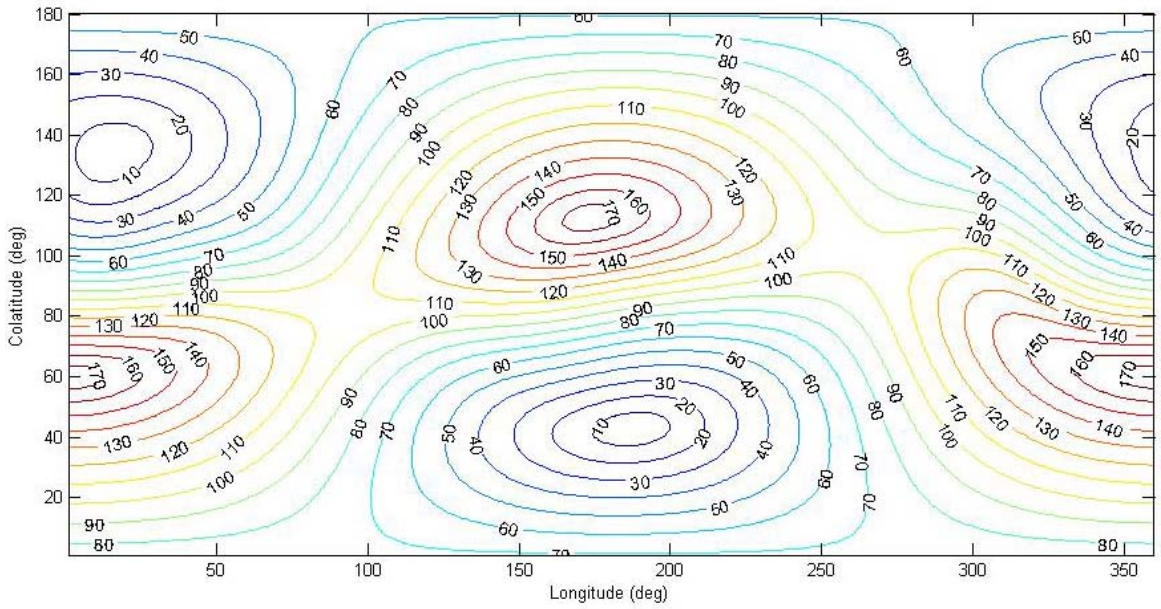

Figure 9 Contour plot of angle between local magnetic field and Sun direction

The peaks and valleys of the contour plot in Figure 9 represent the regions where the spacecraft would be close to a singularity in the Sun-Mag system and therefore may not have the required observations for 3-axis attitude determination.

The geometry of the orbit will determine how frequently the spacecraft is near the singular points. A low inclination orbit or a dawn/dusk Sun synchronous orbit may never approach the singular points; a medium inclination orbit may, at times, be oriented such that it passes through multiple singularities repeatedly. Interestingly, sample data for CP3's orbit does not travel near the singular points; that is, the angle between the Sun and the local magnetic field is always over 20 degrees. However, GeneSat's orbit—a good example of a CubeSat in a medium inclination LEO—does travel near the singular points. Figure 10 shows the angle between the local magnetic field and the Sun direction over 6 months of GeneSat's orbit.

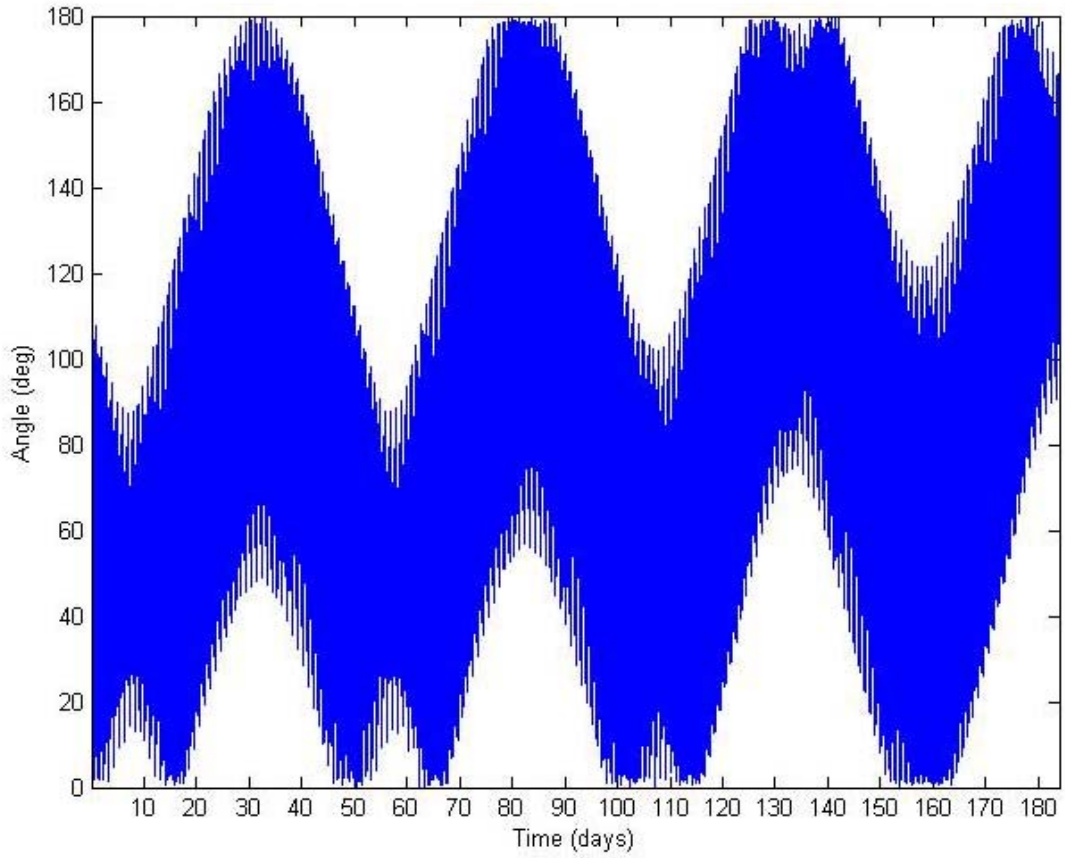

Figure 10 Angle between local magnetic field and Sun direction, GeneSat simulation

In the case depicted in Figure 10, the Sun-Mag vectors are within 10 degrees of each other 1.88% of the time and within 20 degrees 7.42% of the time. Note, the close proximity to the singularities is not evenly distributed across time; slow changes in the orbit drive the spacecraft closer and further from different singular points. By predicting this on the ground, tests could be scheduled to avoid singularities if needed; however, a wait of several days may be needed.

Ultimately, the size of the regions surrounding the singular points which yield unreliable data is dependent on the accuracy of the attitude sensors. For ideal—or extremely accurate—sensors the region of poor data is very small; however, if the sensors are inaccurate, the region of poor data grows. For example, if both the Sun direction vector and the local magnetic field vector readings are accurate to ±5° and the angle between readings is 10°, the vectors could be parallel or 20°

apart; therefore, data where the vectors are within 10° would be useless. Referring back to the contour plot in Figure 9, it is easy to see how the regions where the data may be corrupted by the singular case grow as the sensor accuracy decreases.

The singularities associated with the Sun-Mag system are inherent in any two vector system. Solving this problem is not the focus of this work; however, several obvious solutions exist and are applicable here. The simplest solution is to ignore the singularity and assume that imperfect sensors are perfect; doing this will result in significantly increased error near the singular points. Alternately, the proximity to the singular points could be checked by checking the angle between the two vectors; a tolerance could be set—depending on the accuracy of the sensors—below which the vectors are treated as potentially parallel. While the spacecraft is in these regions of potentially parallel vectors, several things can be done: provide no attitude data as to prevent providing bad data, assume there is no rotation and use the most recent "good" attitude value, calculate a rotation rate based on recent data and extrapolate, use a one vector method such as a Kalman filter, or use gyroscopes. There are surely other solutions, and the ones suggested will not be evaluated for use in this application; however, there are clearly solutions which exist to this problem.

5.2 Test cases

There are an infinite number of configurations that could be used as far as sensor accuracy, magnetic field model fidelity, propagation duration, and other variables; therefore, a representative group of test cases is outline below. The methods used here can be used to provide a prediction of accuracy once hardware is chosen or when comparing different hardware. Without specific hardware, test cases of various levels are used to provide an idea of the range of possibilities of the system outlined in this work. Sensor errors are modeled as normally distributed angular errors about the "truth" values. This assumes that errors would be normally

distributed, which is not necessarily the case, but provides a good reference. For example, if the accuracy of a certain test case is desired, the sensor errors used provide a good indication of the required sensor accuracy, even if the distribution of the actual sensors is not exactly normally distributed. Using only a normal distribution to model error also assumes there is no constant error. There will inevitably be constant error in the system due to issues like mounting error; however, these should be easy to calibrate on the ground to minimize their impact.

Most of the test cases are run for 2 days and all are at 10 second intervals. The 2 day duration is one of the requirements for the orbit propagator; the 10 second interval provides smooth data without being overly computationally demanding. The LLR magnetic field look-up table is used with sizes ranging from 220 to 20200 data points to cover the reasonable range of the look-up table system. A look-up table with 480 entries is used for the majority of cases, since 480 represents the knee in the curve of accuracy versus size. A test case with no sensor error is also provided for comparison purposes. The Sun sensor is given a smaller error in all of the following test cases; this is because Sun sensors are generally higher accuracy. The variability of the Earth's magnetic field--and the difficulty in modeling it—typically increase the error of magnetic field readings. It is possible to create a system with similar sensor errors; however, this would be an unusual configuration since one sensor would be significantly more complex and expensive. Also, TRIAD lends itself to readings of different error magnitude, so the sensor accuracy imbalance is a benefit when using TRIAD. Note, for this to be the case, measurements must be put into TRIAD in the right order; reversing this results in over 50% increased error for the nominal case.

Table 5 ODAD simulation test cases

Test Case	Duration (days)	Magnetic field look-up table size	Standard deviation of sensor error (deg)	
			Sun	Mag
Very coarse	2	480	10	15
Coarse	2	480	5	10
Nominal	2	480	1	5
Fine	2	480	0.5	3
Very fine	2	480	0.1	1
No sensor error	2	480	0	0
Very coarse (2)	2	220	10	15
No sensor error (2)	2	20200	0	0
Long duration	8	480	1	5

5.3 Results

In this system, there are a series of errors that compound into the total error; Figure 11 on the following page shows all of the errors for the nominal test case. The orbit propagation error is shown on the top and represents the difference between the simplified orbit propagator in Section 3 and SGP4. The magnetic field error is a comparison between the IGRF output for the SGP4 position and look-up table output for the propagator position. The magnetic field sensor error and the Sun sensor error show normally distributed random values that represent sensor errors. The total error is a comparison between the rotation matrix output by the entire system and "true" rotation matrix. The error provided here is the rotation about the eigenaxis of the two rotations. This represents the worst case scenario; depending on the direction, the impact may not be as large as the reported total angular error. The angle between the Sun direction and the local magnetic field is provided along with the colatitudes and longitudes; these provide context for the other plots. Figure 12 shows the first 5 hours of the same data so detail can be seen for several orbits.

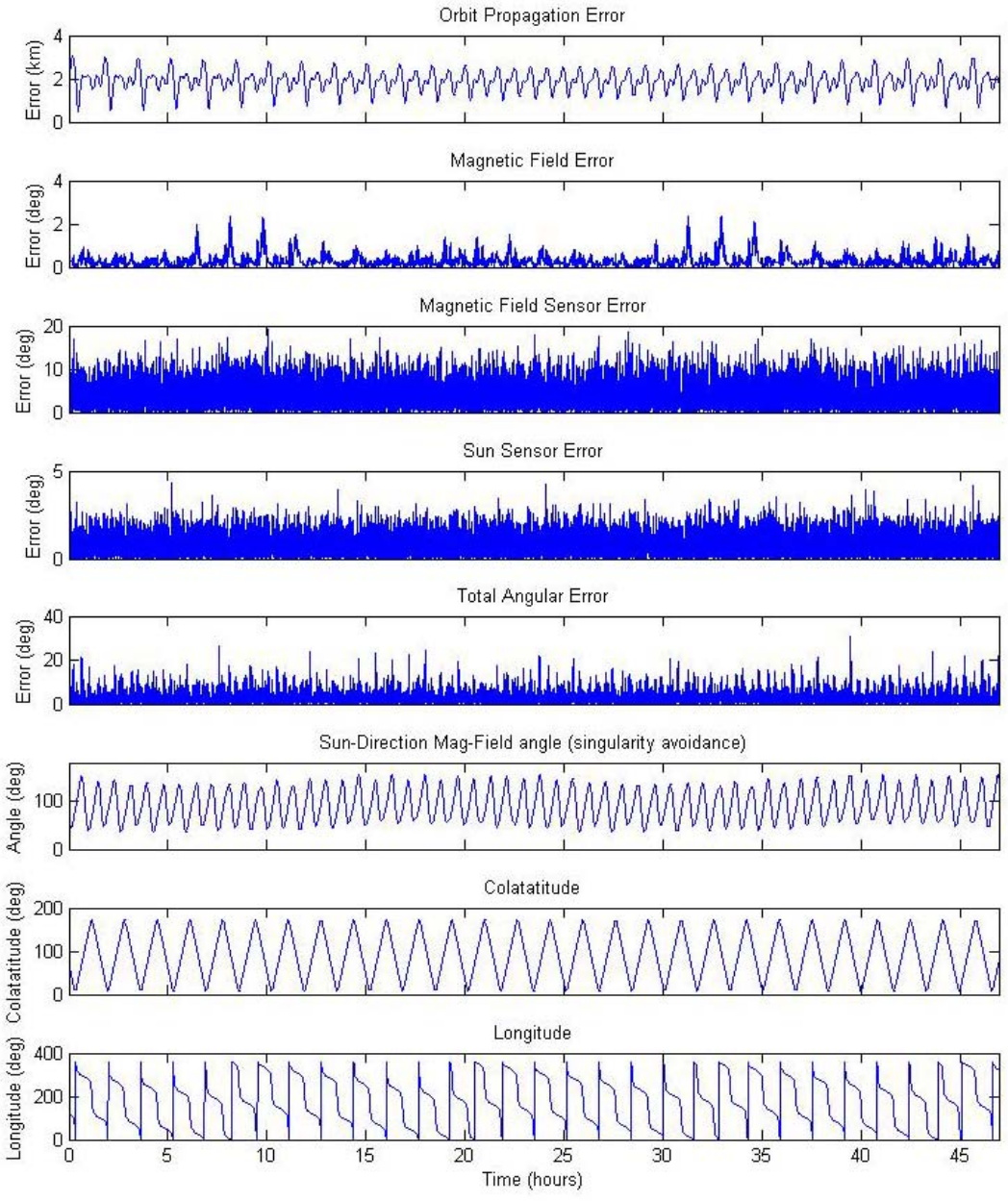

Figure 11 ODAD simulation errors: CP3 orbit, nominal case, 2 days

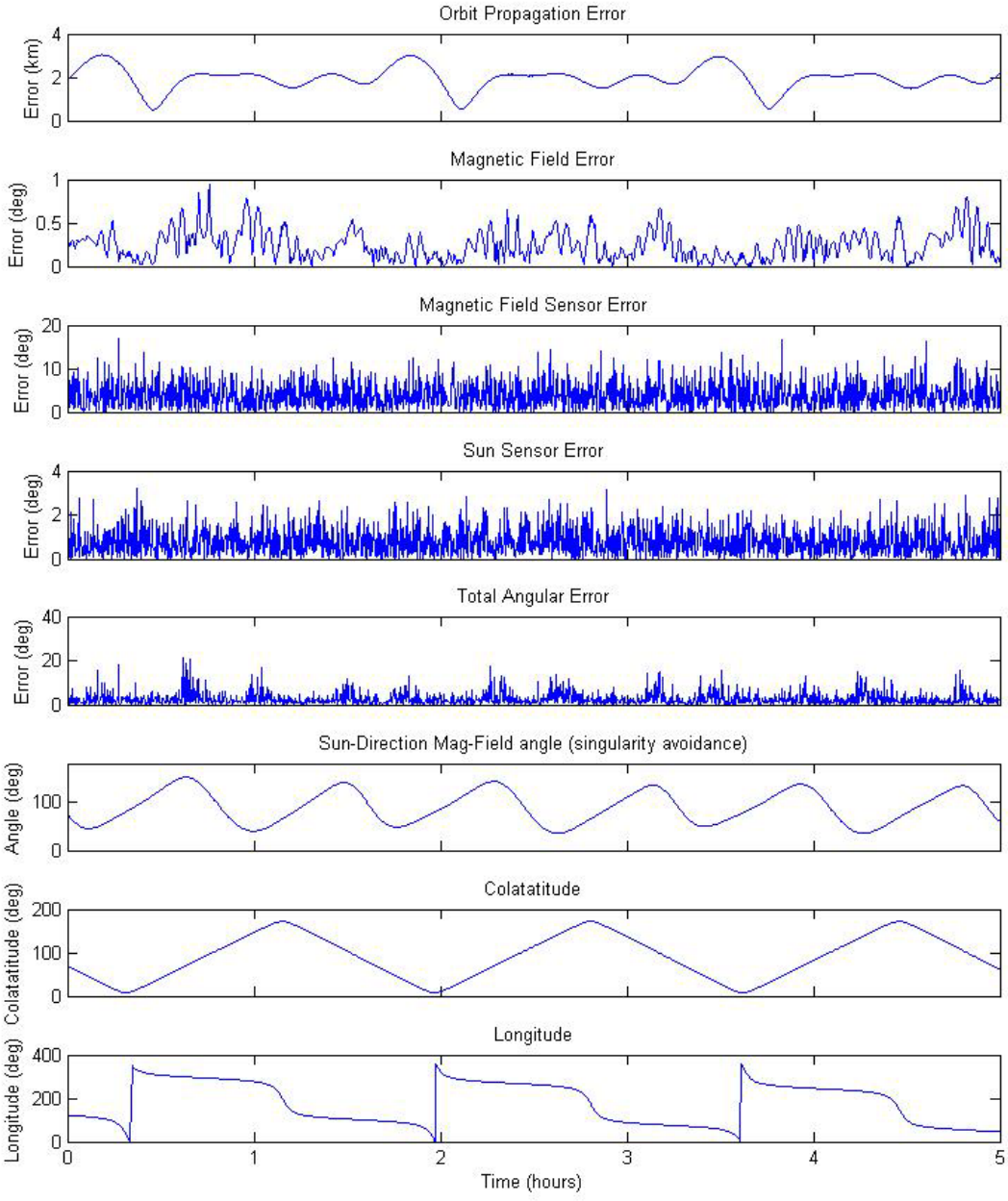

Figure 12 ODAD simulation errors: CP3 orbit, nominal case, 5 hours

The orbit propagation and magnetic field errors in Figure 11 and Figure 12 show similar trends to those discussed earlier. The magnetic field sensor errors and Sun sensor errors are normally distributed random errors of the specified standard deviations. The standard deviation defines the magnitude of the angular errors for each time step; the direction is randomly chosen for each time step. The combination of these errors yields the total angular error plot. The total angular error, with the addition of the sensor errors, is not normally distributed. Figure 13 and Figure 14 are histograms of the total angular error for the no sensor error and nominal test cases respectively. Notice that the combination of normally distributed sensor errors in TRIAD produces a non-normal total error.

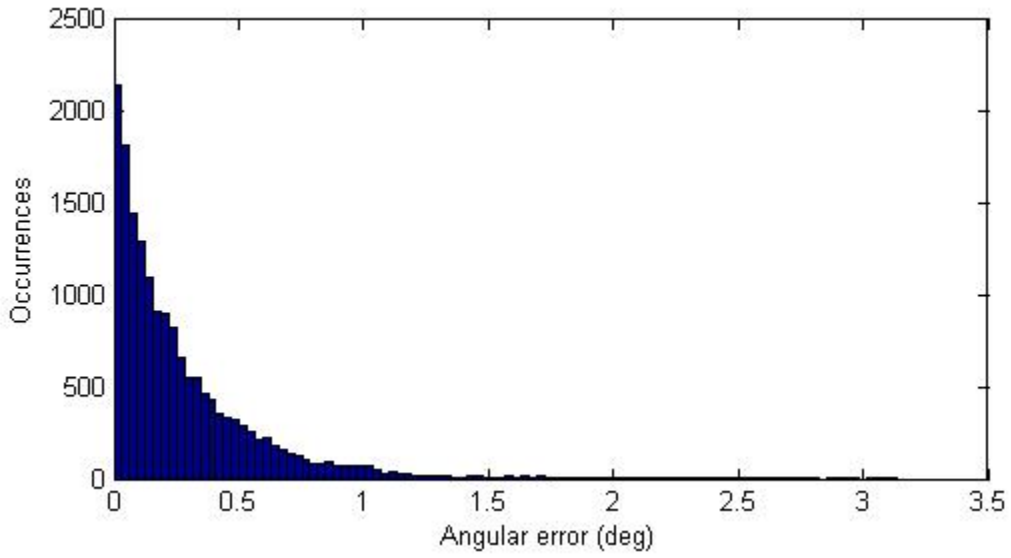

Figure 13 ODAD simulation total error histogram; CP3 orbit, no sensor error, 2 days

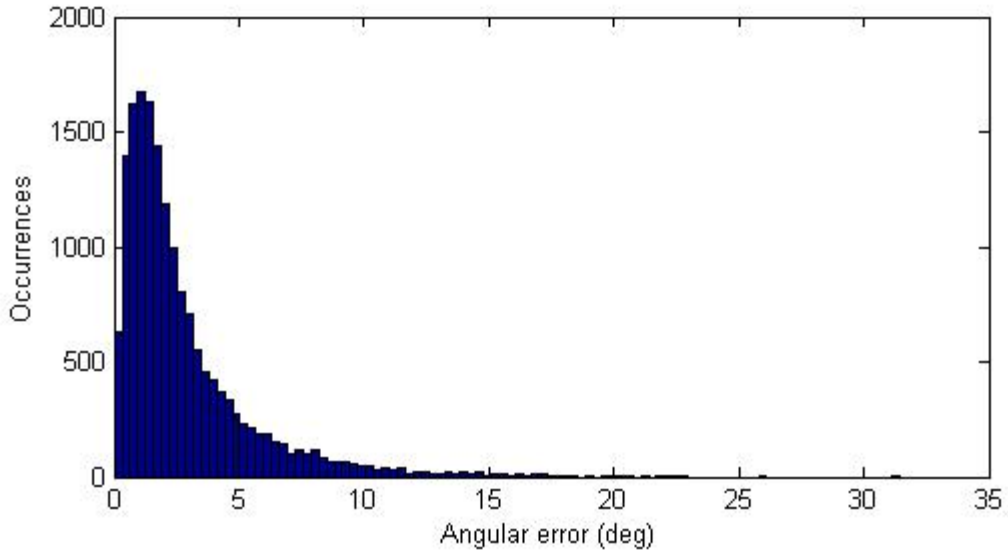

Figure 14 ODAD simulation total error histogram; CP3 orbit, nominal test case, 2 days

The error trends in **Figure 11** and the error distribution in Figure 14 are representative of all test cases—except the no sensor error control—with only changes in amplitude varying between different test cases; therefore, in an effort to avoid redundancy, plots for other data sets are not produced here. Instead, the error of the other test cases is quantified numerically. It is typical to present the first and third standard deviations as a measure of sensor accuracy; however, as is clearly evident in Figure 14, the total angular error is not normally distributed. Therefore, alternate parameters are used to aid in quantifying the error; the mean error, median error and 99^{th} percentile error are presented along with the standard deviation. Table 6 summarizes the results for all of the test cases.

Table 6 ODAD simulation results; CP3, all test cases

Test Case	Duration (days)	Magnetic field look-up table size	Standard deviation of sensor error (deg)		Total error (deg)			
			Sun	Mag	Mean	Median	99th percentile	standard deviation
Very coarse	2	480	10	15	13.02	11.01	48.21	9.75
Coarse	2	480	5	10	7.38	6.06	28.60	5.81
Nominal	2	480	1	5	2.74	1.88	13.04	2.67
Fine	2	480	0.5	3	1.64	1.08	7.94	1.65
Very fine	2	480	0.1	1	0.60	0.40	2.87	0.61
No sensor error	2	480	0	0	0.28	0.18	1.66	0.33
Very coarse (2)	2	220	10	15	13.01	11.01	48.21	9.80
No sensor error (2)	2	20200	0	0	0.09	0.06	0.52	0.10
Long duration	8	480	1	5	2.76	1.88	13.27	2.72

Table 6 shows that the system outlined above is capable of meeting the requirements in Chapter 1. The first section of the table shows that the nominal and better test cases accomplish the target accuracy at least 99% of the time. All but the worst test case meets the mandatory accuracy requirement at least 99% of the time. As expected, there is a clear trend of increased overall accuracy with increased sensor accuracy. The very coarse and very fine test cases represent bounds that a CubeSat mission would probably not achieve; this puts the design space of this system into perspective.

In the second part of the table, the very coarse (2) and no sensor error (2) display the significance of the magnetic field look-up table. Section 3.1 covers the magnetic field model in depth; however, these results add to the ones previously presented. The very coarse (2) results show that

the accuracy of the magnetic field look up table is not important with very crude readings; the results are not significantly different than those produced with a magnetic field look-up table of over twice the size. On the more precise side of things, the no sensor error (2) test case shows upper limit of accuracy in this system; once the sensors have no error, increased accuracy must come from other components of the system. Increasing the size of the magnetic field look-up table significantly increases the accuracy; however, this is still a small change with a very large look-up table.

The third part of the table shows data for 8 days of simulation. The error of the long duration test case is only slightly larger than the nominal case. This is a function of the small increase in orbit propagation error due to the longer propagation time. The increase in error is very small and clearly the system is stable for longer than 2 days.

The nominal test case represents sensor errors that are potentially within the scope of CubeSat mission. This data shows that the ODAD system is capable of providing very accurate attitude measurements; the overall accuracy is strongly dependant on the accuracy of the sensors used and not severely limited by the ODAD system. While complex routines could provide better data, the ODAD system is clearly a reasonable solution for a mission which demands only modest attitude knowledge.

5.4 Un-Modeled Errors

While an attempt was made to make the ODAD system incorporate all potential error, there are some un-modeled errors. These errors are mainly due to hardware and software limitations that are impossible to quantify without porting code and selecting hardware. The following are known sources of error that have not been included in the previous results. Inevitably there will be other sources of error that are neither modeled nor identified here; the ultimate test is to build and fly

the system. A major advantage of CubeSat programs is that new systems can be flown and tested inexpensively and without major risks.

5.4.1 Clock Errors

The orbit propagator fundamentally relies on time to describe the position of the spacecraft. All clocks inherently have some level of error; in this case, the clock error will translate into position error from the orbit propagator. This will affect the magnetic field model accuracy and ultimately the accuracy of the attitude estimate. Modern clocks can be extremely accurate and any clock drift can be corrected by frequent updates; however, clock error of one second would roughly translate into a position error in the velocity vector direction equal to that of the orbital velocity times one second—for LEO this is several kilometers. Therefore, even small errors are significant. The error of setting the clock may even be significant; a more accurate clock update method may be needed.

5.4.2 Fixed Point

The modeling and simulation done in the previous sections were all carried out in MATLAB using floating point numbers and operations. There are small low power floating point processors and fixed point processors with emulation; however, the processors with flight heritage on PolySat projects are fixed point. These processors have been used in the past because they are the smallest and lowest power processors that meet the requirements. Running this system in fixed point with smaller number sizes will inevitable cause some truncation error. This error can be minimized with efficient design based on the bounds of the values; however, some error is inevitable. Ultimately, this error will probably be small and the system could be run on a floating point processor, but the potential for this error exists.

5.4.3 Trigonometric Functions and Square Roots

The fixed point processors mentioned above do not have built in routines for trigonometric functions, square roots or any other operation outside of basic arithmetic. The complex operations were kept to an absolute minimum in this work; however, without simplifying orbits to squares, operations beyond basic arithmetic are inevitable. Many solutions exist, including look up tables and polynomial expansions. As in this entire thesis, a balance between accuracy and complexity will be required; the simpler the solution, the more error. The solutions flown onboard will not and need not be as accurate as the solutions used in the modeling for this work; this will lead to some level of error.

5.4.4 Run Time

The work done in this paper has not accounted for the time it takes to run the algorithms. Some lag will exist between when sensor readings are taken, when orbital position is calculated, and when the final attitude estimate is produced. The calculations required have been kept extremely simple and should have very short run times; however, the run time is finite and the lag will induce error. This error should be small—assuming the system is run in an efficient and streamlined manner—but this should be considered in the hardware selection and flight software.

Chapter 6: Conclusion

This thesis outlined an orbit determination and 3-axis attitude determination system as applicable to 1U CubeSats and other picosatellites. The constraints imposed by the CubeSat form factor led to the need for a simple configuration and relaxed accuracy requirements. To design a system within the very tight mass, volume, and power constraints inherent in the design of a 1U CubeSat, a balance between hardware complexity, software complexity and accuracy was sought.

The proposed solution consists of a simple orbit propagator, magnetometers with a magnetic field look-up table, sun sensors with an analytic sun direction model, and the TRIAD method to combine vector observations into attitude information. In chapter 2, it was shown that an extremely simple orbit propagator based on several frequently updated parameters can provide orbital position data with average and maximum errors—when compared to SGP4—of less than 3.7km and 10.7km for 14 days. Chapter 3 showed that a small look-up table can provide accurate magnetic field data; only 480 data points provide a mean error of approximately 0.2° and a maximum error of approximately 2°—when compared to the IGRF model. Chapter 3 also showed that—as expected—the Sun's location can easily and accurately be predicted for use with Sun sensors.

Combining the magnetic field and Sun direction models with inaccurate sensors and the TRIAD method results in useful attitude information from a very simple system. A system with Sun sensor error standard deviation of 1° and magnetometer error standard deviation of 5° yields results with average error of only 2.74°, and 99% of the errors in this case are less than approximately 13°.

The system outlined provides crude attitude determination with software and hardware requirements that are well within the capabilities of current 1U CubeSats—something that many other systems like extended Kalman filters or star trackers cannot do. It also provides an

excellent starting point for future ADCS systems, which will significantly increase the ability of CubeSats.

6.1 Future Work

This thesis represents a "proof of concept," but is in no way the entire package required to implement an attitude determination system on a CubeSat; there is a significant amount of work that will be required to accomplish this objective.

6.1.1 Code Porting

The system developed in this thesis is written, modeled and simulated in MATLAB. MATLAB has become a standard amongst the aerospace community for many applications; however, it is not applicable to onboard use. The orbit propagator, magnetic field look-up table, Sun direction model, TRIAD method and accompanying functions must be ported to an appropriate language—for the PolySat satellites this is typically C. The underlying method has been demonstrated and the MATLAB code provides a good starting point, as pseudocode if nothing else. However, a significant amount of work is required to develop flight software.

Also, work is needed to run this system on a microcontroller. As mentioned previously, typical CubeSat processors run in fixed point and do not provide operations outside of basic algebra. The system outlined in this thesis requires trigonometric functions as well as square roots; these functions must be implemented—potentially as look-up tables or series expansions—in such a way as to not introduce a significant amount of error or computational burden. Also, the impact of running in fixed point must be evaluated; values must be bounded and sized as to prevent overflow and minimize truncation error.

6.1.2 Hardware Support

This thesis has assumed that a certain amount of hardware is feasible and flight ready. In reality, much of the hardware exists, has been flown, and is thoroughly characterized, but not all. If a dedicated microcontroller is used for the ODAD system, the interface between the ODAD processor and the command and data handling processor must be established. The PolySat CubeSats already operate with multiple microcontrollers; however, design and testing of the interface is still required. I highly recommend considering a dedicated processor for attitude determination and control applications; in the future, when higher accuracy is demanded, the data rates and clock rates will potentially be very different between the attitude determination and control system and other CubeSat systems. For example, a closed loop system with motor controllers for reaction wheels may be implemented; the requirements of this system will lend it to a dedicated processor.

Also, the required vector observations must be provided. In depth study of both flight data from current sensors and capabilities of potential sensors is required. It appears that the requirements of the system outlined in this thesis are within the capability of current CubeSat systems; however, this needs to be verified with flight data or ground testing. This thesis should act as a guide to what sort of accuracy is required from the sensors, and the accompanying code can be used to quantify the expected accuracy with specific sensors.

6.1.3 Concept of Operations

While a concept of operations was presented, more work is need. PolySat CP6 should provide good information as to what uplink frequency and data throughput to assume. The concept of operations will need to be analyzed and designed to maintain orbit propagator accuracy within the limits of the uplink margin. This may include reducing the uplink requirement through an

improved onboard clock, identifying the most important parameters to uplink frequently, or other simplifications.

6.1.3 Flight Testing

The ultimate test of the hypotheses presented in this thesis is flight testing. This system should be flown as a secondary experiment—which is feasible because of its small mass, volume, and power requirements. The actual error should be quantified and compared to the theorized error; the model presented here should then be updated to reflect any missing errors. Finally, if the system performs to the desired accuracy, it should be included as a subsystem of the standard PolySat bus so that it can be implemented in a variety of uses for future missions.

6.1.4 Improvement

This system should be continually improved as the requirements become stricter and the capabilities of CubeSat improve. There are many ways in which this system can be improved, for example: orbit propagator accuracy could be improved with a more complex model, the magnetic field look-up table could be improved in both accuracy and efficiency, sensors which provide readings in eclipse—like Earth sensors or star trackers—could be used to provide attitude knowledge for the entire orbit, sensor readings could be smoothed with filtering techniques, the results from the entire system could be smoothed with either filtering or hardware such as gyroscopes, and more complicated methods for computing vector observations—such as QUEST and other optimal methods—could be used instead of the simple yet crude TRIAD method. Ultimately, this is one of the simplest solutions possible and only serves as a starting point; once a solution is implemented, the potential for improvement is endless.

Appendix A

Much of the work presented in this thesis is dependent on the orbits to which it is applied. The orbit propagator was designed specifically for orbits which have certain dominant errors; the accuracy of the orbit propagator affects the accuracy of the magnetic field model, and in turn the entire system.

A survey of CubeSat orbits was performed to evaluate orbits where the work presented here could be useful, and it was the goal of this work to make it applicable to all reasonable CubeSat orbits. The table below shows a CubeSat from each of the launches carrying CubeSats to date. In the cases where multiple CubeSats were launched, a representative CubeSat was chosen—this is a reasonable approach because the difference in CubeSat orbits within a single launch is minimal.

Launches	date	example sat	apo x peri	e	i	raan
Eurorockot	6/30/2003	XI-IV	830x817	0.0009	98.72	32.3
SSETI Express	10/27/2005	XI-V	707x681	0.0019	98.08	277.22
M-V	2/22/2006	CUTE 1.7 + APD (double)	445x276	0.0124	98.12	172.19
DNEPR	7/26/2006	FAIL				
Minotaur	12/16/2006	GeneSat-1 (triple)	383x379	0.0003	40.03	9.73
DNEPR	4/17/2007	CP3	792x645	0.0103	98.01	83
PSLV	4/28/2008	Compass One	636x613	0.0017	97.96	88.09
TacSat-3	5/19/2009	CP6	459x429	0.0029	40.46	173.52

This table shows that there are two types of orbits that CubeSats have been put into: LEO Sun synchronous, and nearly-circular medium-inclination LEO orbits.

A variety of these orbits were tested with the orbit propagator presented in this work, and they all yielded similar results. For this reason, conciseness was chosen and a few representative orbits were chosen; namely, CP3 and GeneSat because they represented the two general categories of

orbits that CubeSats have been placed in. The two line elements that were used for orbits propagation of CP3 and GeneSat follow.

CP3 TLE:

1 31129U 07012N 09023.12586917 .00000065 00000-0 25904-4 0 6531
2 31129 98.0116 83.0042 0103028 14.9359 345.4849 14.52100354 93830

GeneSat TLE:

1 29655U 06058C 08023.55792021 .00006959 00000-0 10029-3 0 3213
2 29655 040.0257 119.0132 0002790 015.5454 344.5441 15.56156582 62676

These TLEs were propagated forward from 1-23-2008, typically in 10 second increments and for either 2, 8, or 14 days.

References

[1] Larson, Wiley, and James Wertz. Space Mission Analysis and Design. El Segundo, CA: Microcosm , 1999.

[2] Sturm, Erick Jonathan II, "Magnetic Attitude Estimation of a Tumbling Spacecraft." MS thesis Cal Poly San Luis Obispo, 2005. Print.

[3] Brown, Charles D.. Elements of Spacecraft Design. Reston, VA: American Institute of Aeronautics and Astronautics, 2002.

[4] Prussing, John, and Bruce Conway. Orbital Mechanics. New York, NY: Oxford University Press, 1993.

[5] S. Maus, S. Macmillan, T. Chernova, S. Choi, D. Dater, V. Golovkov, V. Lesur, F. Lowes, H. Lühr, W. Mai, S. McLean, N. Olsen, M. Rother, T. Sabaka, A. Thomson and T. Zvereva "10th Generation International Geomagnetic Reference Field". Geophys. J. Int., vol. 161 2005: pp. 561-565.

[6] Wertz, James. Spacecraft Attitude Determination and Control. Dordrecht, The Netherlands: Kluwer, 1978.

[7] Lowes, F. J.. "An estimate of the errors of the IGRF/DGRF fields 1945-200". Earth Planets Space, Vol. 52 2000: pp. 1207-1211.

[8] The Astronomical Almanac. Washington, DC: U.S Government Printing Office, 2005.

[9] Black, Harold. "A Passive System for Determining the Attitude of a Satellite". AIAA Journal, Vol. 2, No. 7 July 1964: pp. 1350-1351.

[10] Wahba, Grace. "Problem 65-1: A Least Squares Estimate of Satellite Attitude". SIAM Review July 1965: p. 409.

[11] Shuster, M.D. and S.D. Oh, "Three-Axis Attitude Determination from Vector Observations". Journal of Guidance and Control, Vol. 4, No. 1, January-February 1981, pp. 70-77.

[12] Markley F. Landis and Daniele Mortari, "How to Estimate Attitude From Vector Observations", Advances in the Astronautical Sciences, Astrodynamics, Vol. 103, Part III, 1999, pp. 1979-1998.