DEVELOPMENT OF MAGNETOMETER-BASED ORBIT AND ATTITUDE DETERMINATION FOR NANOSATELLITES

THOMAS WRIGHT

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN EARTH AND SPACE SCIENCE
YORK UNIVERSITY, TORONTO, ONTARIO

AUGUST, 2014

Abstract

Attitude and orbit determination are critical parts of nanosatellite mission operations. The ability to perform attitude and orbit determination autonomously could lead to a wider array of mission possibilities for nanosatellites. This research examines the feasibility of using low-cost magnetometer measurements as a method of autonomous, simultaneous orbit and attitude determination for the novel application of redundancy on nanosatellites. Individual Extended Kalman Filters (EKFs) are developed for both attitude determination and orbit determination. Simulations are run to compare the developed systems with previous work on attitude and orbit determination. The EKFs are combined to provide both attitude and orbit determination simultaneously. Simulations are run and show that this approach for autonomous attitude and orbit determination on nanosatellites provides 8.5° and 12.5 km of attitude and orbit knowledge, respectively. The results of the simulations are then validated using Hardware-In-The-Loop (HITL) testing. Additionally, a Helmholtz cage is evaluated for future use in the HITL test setup.

Acknowledgements

I would like to acknowledge my supervisors Professor Sunil Bisnath and Professor Regina Lee for their guidance and support. I will carry the skills they helped me to develop through the rest of my career. I would also like to thank the grad students in both the GNSS and YuSEND Labs for their assistance and encouragement throughout my studies. And thank you to my wife, Janet, the myriad things she has done to help me through my Master's education have been invaluable.

Table of Contents

List of Tables

List of Figures

List of Symbols

Quaternions:

Quaternion	q or q'

Scalars:

Area	a
Drag	$b^{\star}$
Dipole moment	d
Voltage	e
Spherical harmonic coefficients	$g_{j,i}$ and $h_{j,i}$
Order	i
Degree	j
RK4 terms	k_1 to k_4
Length	l
Mean anomaly	m
Demagnetizing factor	n_D
Pole number	$\mathbb{m}$
Legendre function	$p_{j,i}$
Radial distance	r
Earth radius	r_E
Potential function	s
Time	t
Variable	u
Combined spherical harmonic coefficients	$v_{j,i}$ and $w_{j,i}$
Semi-major axis	α
Kronecker delta	δ_i^0
Eccentricity	ε
Colatitude	θ
Inclination	ι
Earth parameter	κ
Permeability	μ
Resistivity	ρ

Resistance per unit length	ϱ
Longitude	ϕ
RAAN	ψ
Argument of perigee	ω

Vectors:

Magnetic field vector	$\vec{b}$
Calibration bias	$\vec{c}$
Position	$\vec{r}$
Velocity	$\dot{\vec{r}}$
Random measurement error	$\vec{v}$
Random state error	$\vec{w}$
State vector	$\vec{x}$
Measurement vector	$\vec{z}$
Torque	$\vec{\tau}$
Angular rates	$\vec{\omega}$

Matrices:

Dynamic model Jacobian	A
Measurement model Jacobian	H
Moment of inertia	I
Kalman gain	K
Covariance	P
Process noise	Q
Measurement noise	R
Rotation matrix	$\mathcal{R}$
Calibration matrix	U

Other:

Dynamic model	$f(x)$
Function	$g(u)$
Measurement model	$h(x)$

List of Acronyms

Two Unit (CubeSat™)	2U
Three Unit (CubeSat™)	3U
Attitude Determination and Control System	ADCS
Anisotropic Magnetoresistive	AMR
Command And Data Handling	C&DH
Canadian Dollars	CAD
Commercial Off The Shelf	COTS
Extended Kalman Filter	EKF
Electrical Power System	EPS
Guidance Navigation And Control	GN&C
Global Navigation Satellite System	GNSS
Global Positioning System	GPS
Hardware In The Loop	HITL
International Geomagnetic Reference Field	IGRF
Multiplicative Extended Kalman Filter	MEKF
North American Aerospace Defence Command	NORAD
On Board Computer	OBC
Simplified General Perturbations 4	SGP4
Two Line Element	TLE

1 Introduction

Attitude and orbit determination are critical parts of satellite mission operations. These operations are typically performed on the class of satellites known as “nanosatellites” with the aid of ground stations. Nanosatellites are defined as satellites with a mass of between 1 kg and 10 kg (Lee et al., 2012) and are described in detail in Section 1.1. Due to their small size, limited numbers of sensors are available on nanosatellites. The ability to perform attitude and orbit determination autonomously could lead to a wider array of mission possibilities for nanosatellites, including those where ground communications are difficult, no longer possible, or otherwise not desired. This research examines the feasibility of using low-cost magnetometer measurements as a method of autonomous, simultaneous orbit and attitude determination for nanosatellites.

The ever present and reasonably well-modelled magnetic field of the Earth can be directly measured on a nanosatellite using a magnetometer. Most nanosatellite missions already use part of a magnetometer's measurements for attitude determination. The remaining work, which is tackled in this thesis, is to use the same magnetometer measurements to determine the orbit of the nanosatellite. By using magnetometers, attitude and orbit information would be provided by a single, autonomous sensor. With only one sensor needed, functionality is maintained even if other sensors, such as rate gyroscopes or Sun sensors, fail. To accomplish the processing of the magnetometer data in this work, two coupled Extended Kalman Filters (EKFs) are used. The EKF allows optimal filtering of data with a Gaussian distribution which changes over time. The approach is also relatively computationally simple, while still being able to handle non-linear models.

This research focusses on the design and evaluation of the orbit and attitude determination algorithm when using typical nanosatellite hardware, consisting of low-cost, Commercial Off The Shelf (COTS) components. The drawbacks of using COTS components on nanosatellites are that: the sensors usually have lower accuracy and higher noise; the computational ability of the satellite is limited; and there is limited space on and within the nanosatellite. The consequence is that, to save space, the magnetometer is typically housed within the satellite, rather than on a boom where stray magnetic fields caused by the satellite itself can be removed in calculation or to a large extent ignored. Only a few specially-designed nanosatellite missions have been able to employ booms, such as CanX-2 (Sarda et al., 2009) and CINEMA (Lin et al., 2012), shown in Figure 1-1. To account for the effects of stray magnetic fields on the data, Hardware In The Loop (HITL) testing is performed using a partial satellite on an air bearing table, which contains several larger sources of stray magnetic fields including magnetorquers and the wiring that feeds other nearby sensors (Abdelrahman and Park, 2011). The applicability of using a three-axis Helmholtz cage to generate on orbit magnetic fields for testing purposes is also examined.

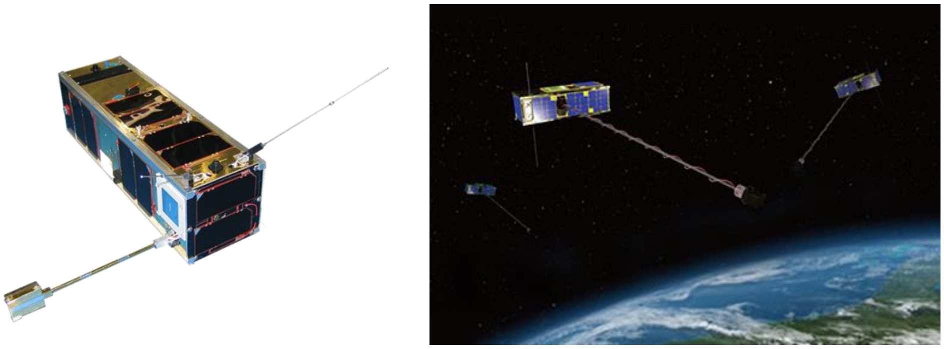

Figure 1-1 CanX-2 (UTIAS SFL, 2014) and CINEMA (Kyung Hee University, 2014) nanosatellites with booms extended.

1.1 Overview of Nanosatellites

One form of nanosatellite that is popular with universities is the CubeSat™, created by California Polytechnic State University along with the CubeSat™ standards. A CubeSat™ is a cube 10 cm to a side with a mass of 1 kg (California Polytechnic State University, 2014). Larger CubeSats™ with 2 or 3 such units attached in a column (2U or 3U) have also flown. Their small size limits the number of components that can be used, but with frequent use of COTS components, build costs are kept low. The use of COTS components also reduces development time by eliminating the need to design many individual components. Given their low costs and fast development times, CubeSats™ are well-suited to technology demonstration missions and for training personnel, particularly at universities.

A typical nanosatellite contains several subsystems including an On Board Computer (OBC), Electrical Power System (EPS), Communications System, Attitude Determination and Control System (ADCS), and a payload. The relation of these systems is shown in Figure 1-2.

The EPS, like all the subsystems, must address the challenges inherent in using significant numbers of COTS components, and being low-mass and low-cost. Due to the small mass budget, a small size is necessary, leading to limited room for solar panels, which limits the power consumption of the electronics in every subsystem. With reduced power generation capabilities and size and mass restrictions, the battery capacity is also small. The EPS must also employ charging circuitry and power regulation that uses space.

The small size and lack of power limit the OBC to smaller and weaker microprocessors and microcontrollers than is otherwise possible, thus limiting the computations that can be performed. To maximize the available computations, the OBC is normally the only microprocessor or microcontroller onboard that can be used for processing. Being the only processor, the OBC must handle all the software tasks that would normally be separated onto other subsystems, including attitude determination and control, and Guidance Navigation and Control (GN&C).

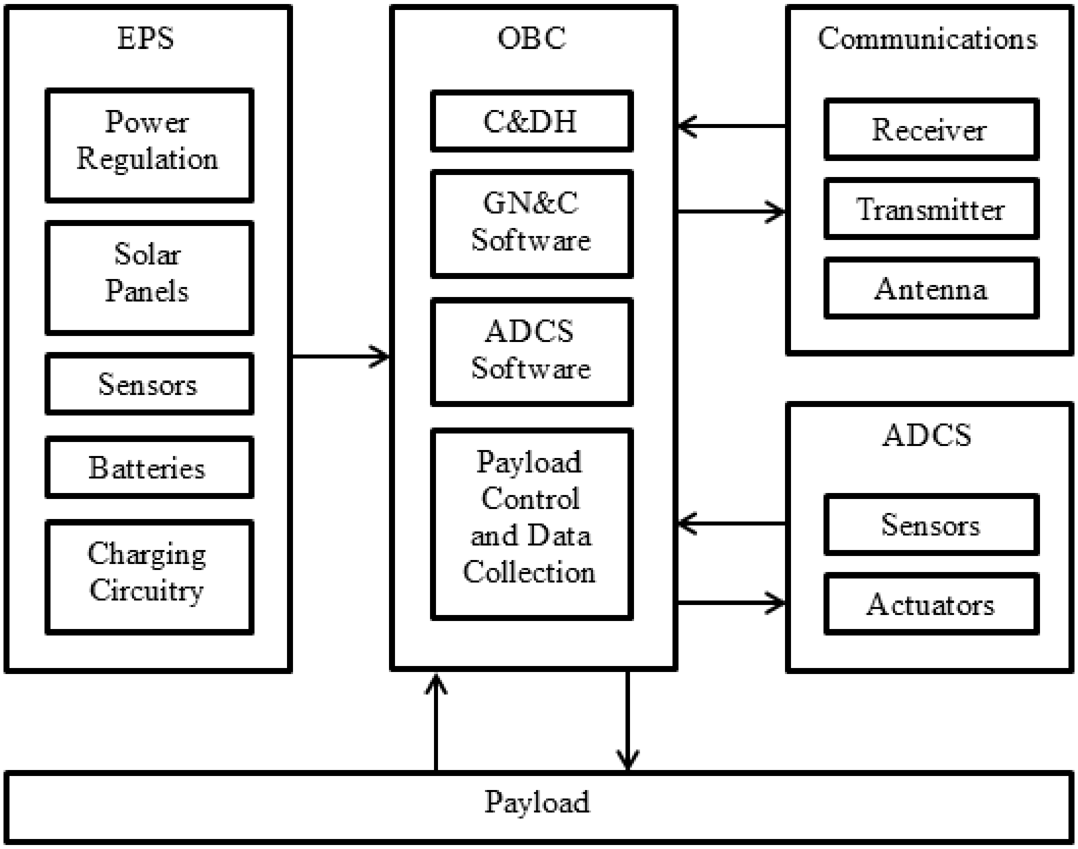

Figure 1-2 Typical nanosatellite subsystem block diagram (Cannata, 2010).

The ADCS often uses passive means to control the satellite, such as gravity gradient or passive magnetic control. Passive methods are advantageous as they require no computing power or electricity to orient the nanosatellite, and require no sensors or actuators. For active control, other methods must be employed, such as spin stabilization, reaction or momentum wheels, or active magnetic control. For attitude determination, several sensors are usually employed, including Sun sensors, horizon sensors, magnetometers, and gyroscopes for angular rates. The ADCS is described in more detail in Section 1.1.

The GN&C system typically consists of orbit propagation using a propagator, such as the Simplified General Perturbations 4 (SGP4), to compute satellite position from ground-based tracking, such as the freely-available Two Line Elements (TLEs) provided by the North American Aerospace Defence Command (NORAD) (Bouwmeester and Guo, 2010). Typically, no system for orbit determination from onboard measurements is included. TLEs are uploaded to the satellite, then propagated using the computationally efficient SGP4 propagator. Alternatively, some nanosatellites use the Global Positioning System (GPS) or another Global Navigation Satellite System (GNSS) for orbit determination, though GNSS receivers are expensive in terms of power and space, and are therefore rarely used. Orbit determination on nanosatellites is examined in more depth in Section 1.1.

As for the other subsystems, communications often use an amateur band receiver and transmitter, and omni-directional antennas due to limited space and pointing accuracy. Payloads are limited or non-existent due to the limited space and power available to them. The purpose of many CubeSats™ is for technology demonstration, and thus the satellite is itself the payload. The structural subsystem on a CubeSat™ is a vibration tolerant box, with rails on four of the edges to allow proper release from the launch vehicle. Vibration tolerance is necessary to ensure it is not damaged by the vibrations of the launch vehicle. There is typically no thermal control subsystem. Temperature of the nanosatellite is controlled passively to reduce size, mass, and power usage.

Typical missions for CubeSats™ include technology demonstration, training, and Earth observation. CubeSats™ are particularly suited to technology demonstration due to their fast completion times and low costs. They are also suitable for the training of personnel due to their inexpensive nature and the use of COTS components. Earth observation is common for CubeSats™, as they generally use lower altitude orbits. Lower orbits are used as these are less expensive to launch to, and they limit the time on orbit if the satellite fails.

1.2 Overview of Attitude Determination on Nanosatellites

Attitude determination is a critical function on satellites as it gives context to measurements and collected data, and is the first step in attitude control. Attitude control allows antennas to be pointed at ground stations for communication, solar panels to be pointed at the Sun for power generation, and instruments to be pointed towards their targets for payload operations. Attitude determination systems are in place on most nanosatellites and utilize a number of sensors to achieve accurate pointing information. Nanosatellites can employ magnetometers, Sun sensors, horizon sensors, rate gyroscopes, and star cameras. Typical examples of these ADCS sensors are shown in Table 1-1.

The most accurate information can be obtained through the use of star cameras and star trackers (Larson and Wertz, 1999), with which attitude knowledge of 3×10^{-5} rad (6 arc sec) has been reported. The advantages of star trackers are high accuracy, the ability to determine attitude in all three axes, and an independence from orbital position. However, due to their large size, very high cost, and high power requirements (Wertz, 1978), they are not suitable for most nanosatellite missions.

Many nanosatellite missions rely on small, low power, low cost, COTS magnetometers, which are accurate to 1.7×10^{-2} rad (1°). Magnetometers are the least accurate and only provide 2 axes of attitude knowledge at any given time. With time and filtering, they can provide 3 axes of attitude knowledge (Gregory, 2001). Their advantages are their small size, low power requirements, low cost, and the ability to take measurements regardless of attitude and orbital position.

Table 1-1 Typical nanosatellite ADCS sensors.

COTS Component (Sensor Type)	Axes	Accuracy	Power (mW)	Cost (CAD)	Notes
Nano Star Tracker™	3	3×10^{-5}rad (6 arc sec)	500	10000	Mass: 350 g Uses custom hardware(Blue Canyon Technologies, 2013)
HMC 5883L (magnetometer)	2	200 nT 2×10^{-2}rad (1°)	0.33	5	Mass: 18 mg (Honeywell International Inc., 2011)
CubeSat Sun sensor	2	9×10^{-3}rad (0.5°)	50	3000	Mass: 5 g (Innovative Solutions in Space, 2014a) Does not work in eclipse
Cubesense (Sun sensor)	2	5×10^{-3} rad (0.3°)	100	6000	Mass: 110 g (Innovative Solutions in Space, 2014b) Both horizon and Sun sensors Doesn't work in eclipse
Cubesense (horizon sensor)	2	3×10^{-3}rad (0.2°)	100	6000	Mass: 110 g (Innovative Solutions in Space, 2014b) Both horizon and Sun sensors
ADXRS453 (rate gyroscope)	1	3×10^{-4}rad/s (0.02°/s)	30	100	Mass: < 5 g Need 3 sensors to get all axes Bias drift rate: 18°/hr/hr (Analog Devices Inc., 2011)

Along with magnetometers, Sun sensors are the most common sensors on nanosatellites (Bouwmeester and Guo, 2010). This is due to their small size, relatively low cost, and direct applicability to Sun pointing. They can be accurate to approximately 5×10^{-3} rad (0.3°) using CMOS sensors. Their disadvantages are that they only provide 2 axes of attitude knowledge and are unable to provide information while the satellite is in eclipse. An alternate method of Sun sensing employed on nanosatellites that reduces power, cost, and space requirements, at the expense of accuracy, is using photodiode arrays. Coarse Sun sensing can also be done using solar panel currents.

Earth sensors, which locate the horizon, can also be used. Earth sensors are directly applicable to attitude determination on Earth observation missions. They are accurate to approximately 3×10^{-3} rad (0.2°) and give 2 axes of attitude information when pointed at the earth. Their main advantage is that the Earth is always visible. Their disadvantages are that they are heavier, more costly, and more power hungry than many of the other sensors.

In addition to having any number of the previously mentioned sensors, rate gyroscopes are commonly included on nanosatellites (Bouwmeester and Guo, 2010) to directly measure the angular rates of the satellite. Though they suffer from systematic errors and cannot determine the attitude of the satellite, they are important for establishing the rate of change of that attitude, which none of the other sensors can do directly.

Magnetometers are one of the most frequently used sensors on nanosatellites, only slightly less so than Sun sensors (Bouwmeester and Guo, 2010). Due to their prevalence, ability to fully determine the attitude over time, observability over the entire orbit, and variability with orbital position, the magnetometer was selected to develop a combined attitude and orbit determination algorithm, and determine its applicability to nanosatellites.

1.3 Overview of Orbit Determination

Orbit determination is necessary for several aspects of satellite operation in general. It provides needed information for the use of attitude determination models, including ephemerides for the Sun and magnetic field models. Knowing the position of the satellite allows for proper understanding of science and payload data by georeferencing the measurements. Position knowledge allows the scheduling of satellite operations, such as: payload measurements, when the desired target is in view; communications, when the ground station is in view; and Sun pointing for increased power generation, when the satellite is not in eclipse.

Several methods exist for orbit determination. Typically, nanosatellites make use of ground tracking by way of NORAD-generated TLEs. TLEs are then uploaded to the satellite and propagated to the current time using the SGP4 orbit propagator (Vallado et al., 2006). Alternatively, propagators such as the Position and Partials as a function of Time (PPT3) or high-quality propagators built on numerical integration of accelerations can be used (Hoots et al., 2004). The SGP4 is advantageous for nanosatellites as it is built to directly propagate the freely-available TLEs. High quality numerical propagators are much more computationally complex, requiring more powerful OBCs. The PPT3 propagator, while also analytical, uses a different formulation of the orbital parameters, which requires additional processing to use. Advantages of using the SGP4 propagator to propagate TLEs are that it is accurate, has a small packet size, uses little computing power, requires no additional hardware, and is a free service. It has the disadvantages of requiring the communication system to be operational for updates and, to keep orbit errors small, requiring updates one or more times per week. The accuracy of this method is approximately 1 km at epoch and loses accuracy at a rate of 1 km per day (Kelso, 2007) for GPS satellites with apogees of approximately 20,000 km, and significantly worse for lower altitude satellites. This degradation in accuracy leads to a need for more frequent TLE updates for lower altitude satellites.

A typical implementation for TLE and SGP4-based orbit determination follows the architecture shown in Figure 1-3 (a). The TLEs, once procured from NORAD, are sent to the satellite from the ground station. They are received by the satellite and processed at various times using the SGP4 propagator on the OBC.

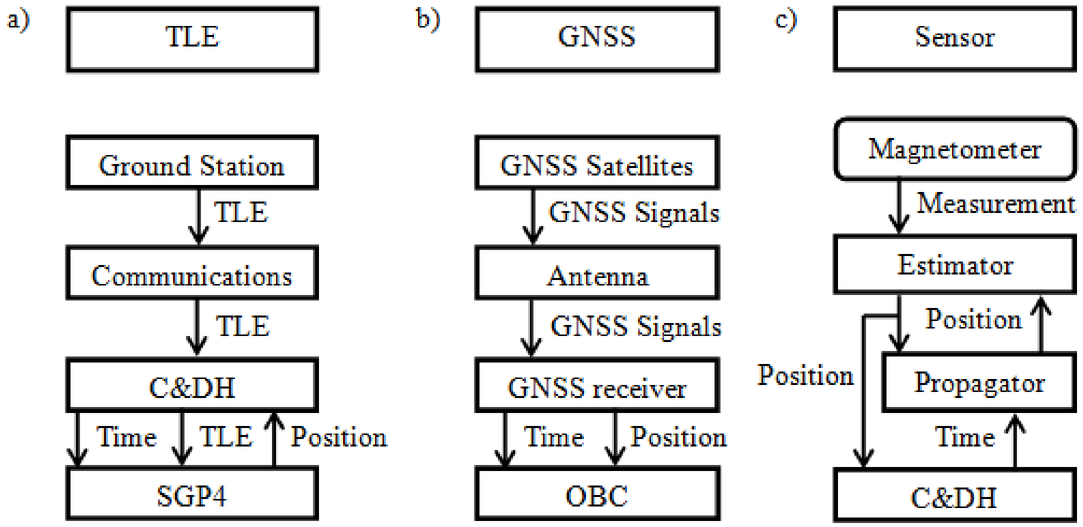

Figure 1-3 Satellite side block diagrams for several orbit determination approaches: a) NORAD based ground tracking, b) GNSS based positioning, c) onboard sensor based orbit determination.

Alternatively, autonomous methods can be employed. The most common method of autonomous orbit determination is by way of GNSS, as on CanX-2 (Greene and Zee, 2009) and Phonesat 2, otherwise known as Alexander (Haque et al., 2013). Signals from GNSS satellites are collected and processed by an onboard GNSS receiver and the position is given directly. Advantages of this method are that it is highly accurate, provides accurate time estimates, and does not require any communications with the ground. However, it has several disadvantages. The necessary components and calculations use significant amounts of power and space, they are more costly, and there can be gaps in the GNSS coverage on orbit. As GNSSs aside from GPS are becoming operational, the gaps in coverage are steadily shrinking, and may soon disappear.

While GNSS can give accuracies of several metres, intermittent use to meet the power requirements on a nanosatellite can reduce the accuracy to 1-2 km, as demonstrated by CanX-2 (Kahr et al., 2010; Greene and Zee, 2009). This accuracy is similar to the accuracy of an up-to-date TLE. Post-processing of GNSS signals on the ground can also give more accurate position information that can be used when analysing payload data. Figure 1-3 (b) shows how GNSS-based autonomous orbit determination requires signals from other satellites to be received and processed by a dedicated subsystem to perform position determination.

Other methods of autonomous orbit determination require the use of a number of sensors, including magnetometers. The advantages of sensor-based orbit determination are that many or all the required sensors may already be on the satellite, requiring little additional space, and no communications with the ground are required. Using nanosatellite quality magnetometers for sensor-based orbit determination has the added benefit of using sensors that are inexpensive compared to GPS. The disadvantages of this method are that it requires significant processing power to run the algorithms and it is less accurate. The method used to perform sensor-based orbit determination is shown in Figure 1-3 (c). Sensor measurements are taken and processed by an estimator and propagator pair such as an EKF. The estimated position measurements are then used by the satellite.

The method examined for this research uses existing hardware to perform autonomous orbit determination, and falls into the category of sensor-based orbit determination. It is advantageous as a backup in the event of failure of the primary orbit determination system, either through failure of a GNSS receiver or the inability to upload replacement TLEs. The accuracy of this method in these circumstances is examined in Chapter 4.

1.4 Overview of the Extended Kalman Filter

The Extended Kalman Filter (EKF) is a non-linear version of the Kalman Filter, which is a least squares estimator for time varying systems. The Kalman Filter is an optimal estimator for dealing with Gaussian systems, where the random variables are normally distributed. Unlike the standard least squares approach, Kalman Filters can be used both for over-determined and under-determined systems. In the case of magnetic attitude and orbit determination, three axes of information are used over time to measure three axes of attitude and their rates, as well as six orbital parameters. The determination of all twelve independent parameters is possible because the EKF starts with an estimate of the previous state and its uncertainty, predicts the current state and its uncertainty, and uses the measurements to refine the predicted current state and its uncertainty. This process is then repeated in a sequential and recursive manner. In order to use the Kalman Filter, which is a linear estimator, on a non-linear system, the first order Taylor series approximation of the non-linear system is taken about the current estimate and used to make the functional model linear at the current epoch.

While working with an under-determined system, it is important to use an estimator, such as the EKF, that can handle the lack of information, essentially turning a time series of measurements into an over-determined system. Of the various estimators, the EKF is the most widely used for real-time attitude estimation (Crassidis et al., 2007). The EKF works best with models that have only small non-linearities. The EKF gives better estimates than simple linear filters, and it has a smaller computational burden than particle filters (Crassidis et al., 2007). Due to these advantages, the EKF is used in this work.

As described by Welch and Bishop (2006), the state vector, $\vec{x}$, comprised of the parameters to be estimated, is propagated by a dynamic model, f. The measurement vector, $\vec{z}$, is related to the state vector by the measurement model, h.

$$\vec{x}_{t+1} = f(\vec{x}_t, \vec{u}, \vec{w}) \tag{1-1}$$

$$\vec{z}_t = h(\vec{x}_t, \vec{v}) \tag{1-2}$$

Where the input vector is $\vec{u}$, the process noise is $\vec{w}$, and measurement noise is $\vec{v}$. The EKF assumes that these noises are independent, zero mean, Gaussian processes that are uncorrelated in time. To apply a least squares estimator to these equations, the covariances of the state vector, P, are also propagated through the linearized models.

To estimate a system, the measurement update is performed first. To perform the measurement update at an epoch t, the a priori measurement, $\vec{z}_t^-$,and Kalman gain, K_t, are calculated from the a priori state vector, $\vec{x}_t^-$, the a priori covariance matrix, P_t^-, the Jacobian of the measurement model, H_t, and the measurement noise, R.

$$\vec{z}_t^- = h(\vec{x}_t^-, 0) \tag{1-3}$$

$$K_t = P_t^- H_t^\top [H_t P_t^- H_t^\top + R]^{-1} \tag{1-4}$$

These are used to estimate the state vector and its covariance matrix.

$$\hat{\vec{x}}_t = \vec{x}_t^- + K_t(\vec{z}_t - \vec{z}_t^-) \tag{1-5}$$

$$\hat{P}_t = P_t^- - K_t H_t P_t^- \tag{1-6}$$

Then the a priori state vector and a priori covariance matrix for the next epoch are calculated from the estimated values, the Jacobian of the dynamic model, A_t, and the process noise matrix, Q:

$$\vec{x}_{t+1}^{-} = f\left(\hat{\vec{x}}_t, \vec{u}, 0\right) \tag{1-7}$$

$$P_{t+1}^{-} = \mathrm{A}_t \hat{P}_t \mathrm{A}_t^{\mathrm{T}} + Q \tag{1-8}$$

The Jacobians A_t and H_t are:

$$\mathrm{A}_t = \left.\frac{\delta f(\vec{x})}{\delta \vec{x}}\right|_{\hat{\vec{x}}_{t-1}} \tag{1-9}$$

$$H_t = \left.\frac{\delta h(\vec{x})}{\delta \vec{x}}\right|_{\vec{x}_t^{-}} \tag{1-10}$$

with equations (1-7) and (1-8) denoting the dynamic update, and equations (1-3) through (1-6) denoting the measurement update, as shown in Figure 1-4. This work assumes the dynamic and measurement models are linear functions of the process and measurement noises. Therefore, the process and measurement noise matrices can be computed beforehand and used for each epoch. These equations are repeated for each epoch to give real-time estimation of the current state vector based on current measurements.

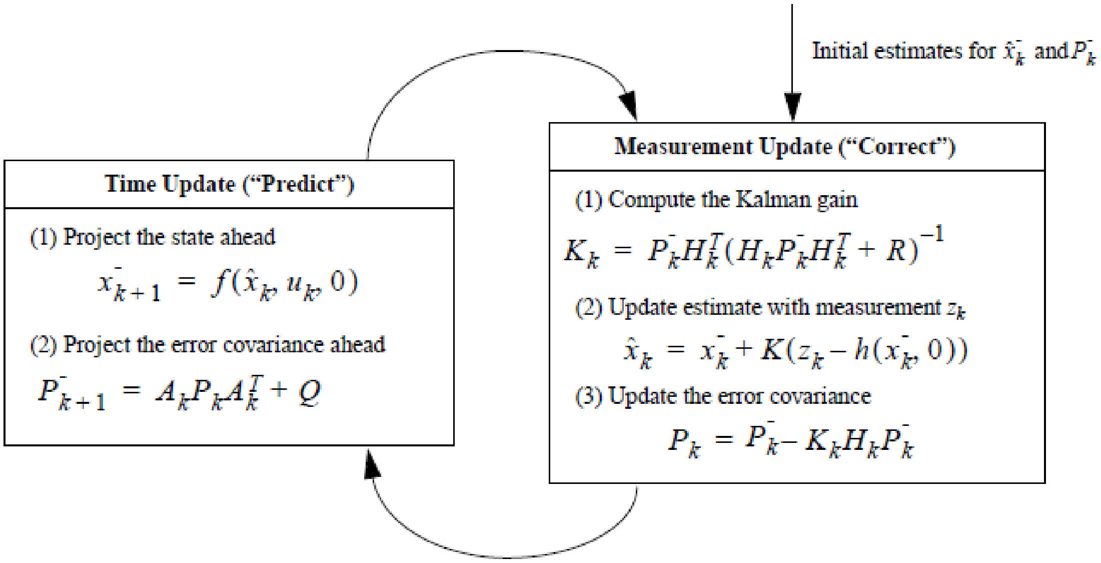
Initial estimates for $\hat{x}_k^-$ and P_k^-
Time Update ("Predict")
(1) Project the state ahead
$x_{k+1}^{-} = f(\hat{x}_k, u_k, 0)$
(2) Project the error covariance ahead
$P_{k+1}^{-} = A_k P_k A_k^T + Q$
Measurement Update ("Correct")
(1) Compute the Kalman gain
$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1}$
(2) Update estimate with measurement z_k
$\hat{x}_k = x_k^- + K(z_k - h(x_k^-, 0))$
(3) Update the error covariance
$P_k = P_k^- - K_k H_k P_k^-$

Figure 1-4 Process diagram outlining the operation of the EKF (Welch and Bishop, 2006)

The specific applications of Extended Kalman Filtering techniques used in this work are described in detail in the individual chapters. Section 3.3 describes the Attitude determination EKF, Section 4.2 describes the Orbit determination EKF, and Section 5.1 describes the interface between the two EKFs when estimating attitude and orbit simultaneously.

1.5 Recent Related Research

Ongoing research is being conducted on various methods of orbit determination using natural phenomena such as the Earth's magnetic field and the positions of various celestial bodies. Using combined magnetometer and Sun sensor data, Psiaki (1999) and Jung and Psiaki (2002) are able to achieve 2 km accuracy in orbit determination. His method involves using high quality sensors, accurate to 10 nT and 0.005°, respectively, and then using the angle between the Sun and the magnetic field to improve the accuracy of the International Geomagnetic Reference Field (IGRF) model. They estimate the Keplerian orbital parameters by using a batch filter to process the data, and a physics-based Keplerian model, including a drag term, for propagating the orbit. This method shows significant improvement at the end of the IGRF's life, but little at the beginning. This process uses higher accuracy sensors than those used on nanosatellites and has difficulty as altitudes rise, and inclinations dip below 45°.

To show applicability to satellites with low computing power, Laneve and Curti (1997) use all of the magnetic field measurements from an entire orbit simultaneously to estimate the orbital parameters. Only the magnitudes of the magnetic field measurements are used by means of a Multi-rate EKF. And they estimate the six classical Keplerian elements and account for only the J2 harmonic perturbation in their propagation of the state. The orbital parameters are estimated to better than 0.01 rad in each parameter and less than 1 km in the semi-major axis using a magnetometer accurate to 0.25 nT. The solution is shown to converge in approximately 15 orbits, or 1 day.

Gregory (2001) and Psiaki et al. (1990) use magnetometers alone to estimate attitude. Using sensors accurate to 500 nT, as is commonly found on nanosatellites, they estimate attitude to an accuracy of 1° in all three axes. They use an EKF to process the data, comparing it directly to the IGRF model. And they use the quaternion, body rates, and drag coefficient as their state variables and propagate them using quaternion kinematics and the Newton-Euler equation, including several torque terms.

Abdelrahman and Park (2011) use only magnetic field measurements for both orbit determination and attitude determination. They are able to achieve 8 km accuracy for orbit determination and 5° and 0.03°/s accuracy for attitude, using sensors from larger satellites which are accurate to 50 nT. EKFs are used to process these data and investigate the differences between a filter that simultaneously estimates both attitude and orbit, and one that estimates them in separate, coupled filters running concurrently. The magnetometer data and a measurement of the derivative of the magnetic field, which they derive numerically from a set of magnetic field measurements, are directly compared with the IGRF model. Accelerations in Cartesian coordinates are used to propagate the orbit and quaternion kinematics and the Newton-Euler equation to propagate the attitude. They estimate the body rates, attitude quaternion, position, velocity, residual magnetic dipole, and drag coefficient as the state variables.

Extrapolating the accuracy of the orbit determination from previous work to magnetometers employed on nanosatellites, accurate to 200 nT, gives an expected positional accuracy of approximately 40 km. From the work done by Laneve and Curti (1997), the orbit solution is expected to converge after approximately 15 orbits. Based on the work of Gregory (2001), Psiaki et al. (1990), and Abdelrahman and Park (2011), the attitude estimation is expected to be around 5° or better and degrade by several degrees when performed simultaneously with orbit determination. Abdelrahman and Park (2011) also show that little accuracy is lost by using two EKFs simultaneously while large amounts of computing power are saved. Previous results using magnetometers are not as accurate as those using recent TLEs at 1 km, the standard approach for CubeSats™, or as GPS solutions which can be better than 5 m, but it has the advantage of being autonomous and requiring very little power.

1.6 Research Contributions

While other attempts are being made to estimate attitude and orbit using magnetometers, they are limited to high accuracy sensors and often use other sensors to supplement the magnetic field measurements in an attempt to improve accuracy. This research attempts to determine the applicability of using only magnetometers for both attitude and orbit determination for the novel application of redundancy on nanosatellites, which use low-grade COTS components.

In this research, two coupled EKFs are developed to estimate the orbit and attitude of a typical nanosatellite. One EKF is developed to estimate the Brouwer mean orbital elements as used by NORAD in supplying the TLEs. The mean orbital elements are propagated using a modified SGP4 propagator in two parts: one part propagates the mean elements; and the other converts them to position and velocity. The second EKF is developed to estimate the attitude quaternion and angular rates, which are propagated using the Newton-Euler equation and quaternion kinematics equations.

The results of this investigation are intended to provide nanosatellites with an alternate approach to orbit determination in case of failure or temporary disablement of the primary orbit determination or attitude determination systems. The thesis will also examine the extent to which magnetometer measurements can be used on nanosatellites. By using onboard methods to perform orbit determination, nanosatellites can be given autonomy and do not need to rely on ground communications to complete their missions. Autonomy will be important in future missions when nanosatellites are used as a cheap alternative for exploring other planets and moons where ground stations and GNSS do not exist. Using functions that are already in use on nanosatellites, namely the SGP4 propagator and attitude propagation equations, adds to the modularity of the system. Computing requirements are restrained by using an analytical orbit propagator, by separating the attitude and orbit determination into two separate, but coupled, EKFs, and due to the relatively low computing requirements of the EKF. The thesis builds on the previous body of work by applying the techniques of magnetometer-based attitude and orbit determination developed for larger satellites to nanosatellites and evaluating the resultant EKFs using HITL testing.

The content of the thesis is as follows. In Chapter 2, background on the magnetic field and the particular model used in the research is given. In Chapter 3, the EKF for attitude determination is examined. In Chapter 4, the EKF for orbit determination is examined. Chapters 3 and 4 provide a baseline from which Chapter 5 examines the effects of running the two EKFs simultaneously. In Chapter 6, a HITL test-bed is introduced and used to test the EKF with the noise environment seen on a nanosatellite. And a summary of the results and avenues of future study are presented in Chapter 7.

2 Earth’s Magnetic Field

The Earth's magnetic field is one of the oldest tools for navigation, having been used in China for nearly 1000 years (Malin, 1987). Originally, the Earth was thought to act much like a large bar magnet, but the magnetic field has been found to be much more complex. Not only does the direction and magnitude of the magnetic field vary with geographical location, in comparison to that of a bar magnet, it also varies with time and solar activity (Stern, 1989).

Currently, the best explanation for the magnetic field is the geodynamo model. The geodynamo model describes a mechanism similar to an electrical generator, often called a dynamo, in which a rotating permanent magnet causes a current to flow in the surrounding coils made of electrically conductive material. In the geodynamo, iron precipitates out of the molten outer core onto the solid inner core at the red boundary in Figure 2-1. As the iron cools, it becomes magnetized, causing the inner core to act as a permanent magnet. With the iron gone, the leftover material in the molten outer core is more buoyant and rises, following a helical motion due to the differences in spin axes and rates between the crust and the inner core as shown in yellow in Figure 2-1. As it rises, it carries charged particles with it, generating a magnetic field. This magnetic field, along with the magnetization of the inner core, forms the bulk of the Earth's main field. Smaller variations near the Earth's surface are also caused by magnetic minerals in the Earth's crust (Roberts and Glatzmaier, 2000).

In addition to the Earth's main field, charged particle motion in the vicinity of Earth also generates a magnetic field. This charged particle motion is responsible for less than 1% (Stern, 1989) of the magnetic field at sea level on days without solar flare activity. Decreasing proportions of the magnetic field are created by internal terrestrial sources as distance from the

Earth increases. During magnetic storms, external sources can be responsible for a much greater percentage of the magnetic field. These storms can affect the magnetic field for several days (Stern, 1989).

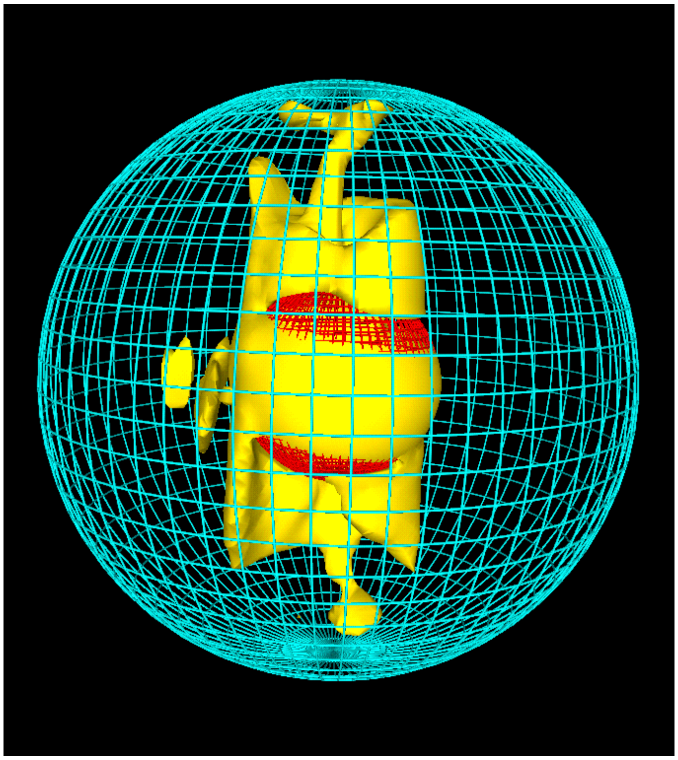

Figure 2-1 The fluid motion in the outer core driving the geodynamo. Yellow areas have the highest velocity. Red denotes the inner core boundary. Blue denotes the interface between the outer core and the mantle. (Glatzmaier, 2010)

2.1 Previous Missions Interacting with the Earth's Magnetic Field

All satellites interact with the Earth's magnetic field in some way, either by using it to determine attitude or orbit, or to change attitude, or to measure the magnetic field and particles influenced by it, or by simply having their attitude perturbed by it.

2.1.1 Missions to Measure the Earth's Magnetic Field

Before satellites, the Earth's main field was well understood, but the external sources were still poorly modelled (Stern, 1989). This limited understanding was sufficient for use on the ground and for low orbiting satellites, but with the advent of the space industry, it was found that a lack of understanding of the external field caused measurements in higher orbits to be modelled less accurately. So new models were needed that could more accurately describe the environment in space. In order to do so, missions like MAGSAT, Ørsted, CHAMP, and SAC-C (Friis-Christensen et al., 2006) were devised to provide in-situ measurements. Missions of this kind needed to have high accuracy magnetometers and often employed both types: total field magnetometers to measure the magnitude of the magnetic field; and 3-axis vector magnetometers to measure the components of the magnetic field in three axes. The magnetometers were isolated on long booms to reduce the effect of stray magnetic fields from the satellites themselves, and pains were taken in the design phase to reduce stray magnetic fields generated by current loops (Acuña, 2002). High accuracy orbit and attitude determination, combined with a polar orbit that allowed measurement over the entire Earth at various times of day, facilitated matching the measurements to specific locations at specific times. These measurements made possible a much greater understanding of the workings of the external magnetic field sources, and in turn allowed for separation of the models for the internal and external fields (Stern, 1989).

2.1.2 Missions that Utilize the Earth's Magnetic Field

Most nanosatellites use the Earth's magnetic field, with nearly 30% utilizing magnetometers, and nearly two thirds utilizing magnetorquers or passive magnetic control (Bouwmeester and Guo, 2010). For example, ION, as shown in Gregory (2001), uses the magnetic field for both attitude determination and attitude control. Attitude is determined with an EKF processing both magnetic field and gyroscope measurements. Attitude control is performed using only magnetorquers in a linear quadratic regulator control algorithm. While attitude control can directly use measurements of the magnetic field to determine attitude, as with an EKF, it requires that the measurements be compared with their expected values. These expected values are obtained from models of the Earth’s magnetic field. The models are stored and evaluated on the satellite to allow greater flexibility in terms of satellite position, and to minimize use of the communications uplink.

For attitude determination, the expected attitude is used to generate a magnetic field vector in the satellite body frame. This expected magnetic field vector is generated from a model of the Earth’s magnetic field at the satellite’s position, and rotated into the body frame using the expected attitude. This expected magnetic field vector is then compared to the measurement made, and the difference is used to correct the expected satellite position.

For active magnetic attitude control, like that on ION (Gregory, 2001), the difference between the current attitude and the desired attitude is used to calculate the torque to be applied by magnetic coils. The difference between the magnetic field and the magnetic dipole created by the coils generates a torque on the satellite. However, no torque can be applied along the magnetic field vector (Silani and Lovera, 2005). The controller, like the attitude determination algorithm, must make use of the changing nature of the magnetic field for full 3-axis control.

For passive magnetic attitude control, like that used on QuakeSat (Long et al., 2002), the satellite orients itself in line with the magnetic field. The control torques in passive magnetic control are generated by the same method as those for active magnetic control except the dipole is constant. The dipole is selected before launch and permanent magnets are affixed to the satellite to generate the control torques. In order to reduce oscillations about the magnetic field, dampers are required. The stability of the system can be investigated before launch through the use of simulations with magnetic field vectors generated by models of the Earth’s magnetic field.

2.1.3 Magnetometers Utilized on Satellites

Magnetometers are used to measure magnetic fields. They are unable to differentiate between the sources of the measured field. There are two groups of magnetometers: those that measure magnitude only, and those that measure the magnetic field in a particular direction. The latter, vector magnetometers, are used to measure the magnetic field in all three axes by combining three orthogonal sensing modules. Both kinds are used on satellites, but only vector magnetometers can be used for attitude determination, as total field magnetometers make no distinction of the direction of the magnetic field. Of the many types of magnetometers, most use too much power to function on a nanosatellite, are not small enough to be used on nanosatellites, or are not sensitive enough to measure the Earth's magnetic field. Fluxgate, and anisotropic magnetoresistive (AMR) magnetometers are the only ones capable of general use on nanosatellites, though optically pumped magnetometers show promise (Lenz and Edelstein, 2006).

The total field magnetometer that requires the least power and volume is the optically pumped magnetometer. Optically pumped magnetometers have flight heritage on missions like the OGO satellites and MAGSAT (Mobley et al., 1980). With appropriate readout electronics, they can measure magnetic fields as small as 10^{-3} nT. They suffer from a dead zone, where at certain attitudes the magnetic field cannot be measured, so multiple magnetometers are needed to guarantee useable measurements. Traditionally, they are large chambers filled with cesium or helium combined with a variable frequency RF field generator (Lenz and Edelstein, 2006). There have been attempts to miniaturize these chambers in the past decade (Schwindt et al., 2004). As of yet, there are no commercially available miniature optically pumped magnetometers, so they still remain unsuitable in nanosatellite applications.

Fluxgate magnetometers use the change of permeability in a magnetic core between its saturated and unsaturated states to measure the strength of a magnetic field. The core is wound with two coils, a drive coil and a response coil. The drive coil is driven with a strong AC voltage that saturates the core in alternating directions. The permeability of the core changes from high values when it is unsaturated to low values when it is saturated. This causes a response at twice the frequency of the drive signal. The response coil output is filtered to remove the drive signal and other unwanted harmonics, retaining only the desired response frequency (Lenz and Edelstein, 2006). Since the applied magnetic field is always in the same direction, the bias of the AC response at twice the drive frequency is proportional to the strength of one axis of the applied magnetic field. Fluxgate magnetometers cost $50 or more and can require significant power to run, from 180 mW (Fat Quarters Software and Electronics, 2012) to as much as 1 W. They can be relatively small, requiring 7 cm^3 or more of room for all 3 axes. Their main advantage is their precision, being able to measure magnetic fields between 10^{-2} and 10^7 nT to a precision of 10^{-2} nT (Lenz and Edelstein, 2006).

AMR sensors use a change in the resistance of the sensor element with the applied magnetic field to measure that magnetic field. The resistance of AMR sensing elements changes based on the angle between the magnetization of the element and the direction of current flow. The direction of magnetization of the sensing element is proportional to the applied magnetic field. This behaviour is strongly exhibited by the alloy commonly called permalloy, so it is widely used to make sensing elements (Lenz and Edelstein, 2006). The magnetization of the sensing element tends to stay within the plane of a thin film, and along the length of the sensing element. As a magnetic field is applied in the plane of the film and perpendicular to its length, the magnetization of the sensing element rotates toward the applied field. The amount of rotation is proportional to the strength of the applied field (Lenz and Edelstein, 2006). In order to make the resistance a linear function of the strength of the applied field, the current needs to flow at 45° to the magnetization at zero external field. To make the current flow at 45° to the magnetization, which lies along the length of the element, angled paths similar to the stripes on a barber pole are added to the sensing element, as seen in Figure 2-2 (Caruso et al., 1998). To decrease power consumption and increase sensitivity, the elements follow a long, serpentine path. As the response of the element is fairly weak, several elements are placed in a bridge configuration, amplifying their responses and allowing the change to be measured as a change in voltage across the two legs of the bridge (Lenz and Edelstein, 2006).

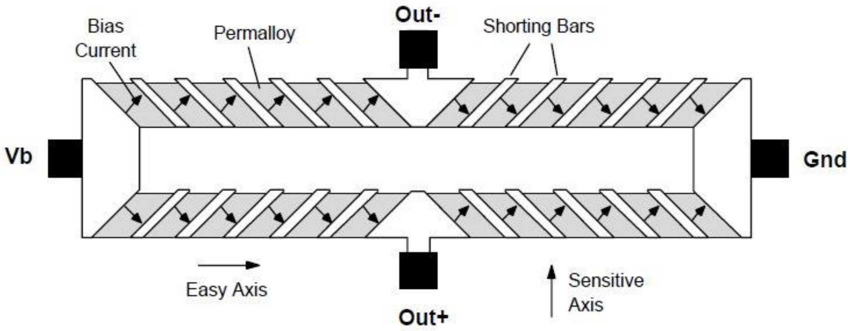

Figure 2-2. Operation of AMR magnetic sensing elements with barber pole bias (Caruso et al., 1998).

AMR magnetometers are able to measure between 1 and 10^5 nT to a resolution of 10^2 nT (Honeywell International Inc., 2011; Caruso et al., 1998). These sensors can measure static fields and those fluctuating at less than 1 GHz (Lenz and Edelstein, 2006). AMR sensors suffer from errors due to mismatched sensor element resistances, which can be compensated for by taking two measurements with the sensing element at opposite magnetizations. The two measurements give opposite values which both suffer from the same bias. Subtracting the measurements from each other eliminates the inherent resistance errors and their associated noise (Lenz and

Edelstein, 2006). In quickly changing fields, taking two measurements can add an error due to the change in the magnetic field between the two measurements. This method of measuring reduces the response time and increases power consumption as it takes time and energy to set the magnetization for each measurement.

AMR sensors can be as small as a few millimetres per side, and require little power, using between 0.1 and 0.5 mW. They have a wide operating temperature range, which can be from -50 to 200 °C (Lenz and Edelstein, 2006; Honeywell International Inc., 2011), and cost approximately $5 for a three-axis sensor (Digi-Key Corporation, 2012).

Despite their lower sensitivity, the low-cost, low power requirements, and small size of AMR magnetometers make them desirable for general purpose use on nanosatellites. There are several sources of noise and bias in measurements taken with AMR magnetometers. Nearby stray magnetic fields, caused by hard (permanent magnets), soft (ferromagnetic), or electrical sources, sensor misalignment, temperature effects, and noise in the readout electronics all affect the accuracy of the measurements (Acuña, 2002; Caruso et al., 1998).

When measuring magnetic fields, nearby permanent magnets can introduce errors into the measurements. These errors are constant biases that depend on the relative positioning of the permanent magnet and the magnetometer. In the case of dipoles, these can be calculated using two measurements at different distances. The dipole can then be removed from the measurement. Alternatively, the permanent magnet can be removed. In the case of spacecraft with temperature stable permanent magnets, the dipole's size and location can be measured before the spacecraft is launched and subtracted from the measurements (Acuña, 2002).

Soft magnetic disturbances are caused by nearby magnetic material, which take on the ambient magnetic field and amplify it. These errors depend both on the ambient magnetic field and the distance between the soft magnet and the magnetometer. Like hard magnetic disturbances, they can be calculated and removed by using two magnetometers, or removing the source. For spacecraft measurements, these sources are removed or replaced with non-magnetic material as much as possible (Acuña, 2002).

Nearby electrical systems and currents can create magnetic fields that add errors to the desired measurement. To reduce these errors, several wiring and layout techniques can be employed. For large loads, using twisted pair wiring reduces the stray magnetic fields. For power systems, using single point grounding reduces stray fields. To compensate for circulating currents, dummy loops can be used to reduce the generated fields. In North American terrestrial applications, such as magnetic observatories and laboratory facilities, fields from power lines can be removed by rejecting fields at 50 or 60 Hz and their harmonics. As with other external sources, electrical magnetic sources can be placed at a distance and the main dipole field removed using two magnetometers. In some cases, large sources of electrical magnetic disturbances can be removed by turning off the source while making measurements. Often, for space applications, large booms are employed to minimize the effect of external magnetic sources in the satellite. As well, two magnetometers at different distances on the boom can be used to calculate the main dipole effect from the satellite, and remove it from measurements. Magnetic sources with $2\mathbb{m}$ poles produce fields that vary with $r^{-(2+\mathbb{m})}$. Only the main dipole ($\mathbb{m} = 1$) is calculated as quadrupole ($\mathbb{m} = 2$) and higher order pole arrangements produce magnetic fields that quickly die off (Acuña, 2002).

As the temperature of the magnetometer changes, the properties of the sensing elements change. To compensate for the changes, the magnetometer can be calibrated across a range of temperatures in a known static magnetic field. As most magnetic sensors operate in a linear range, a simple scale factor is all that is needed. This scale factor is then applied directly to the measurements (Honeywell International Inc., 2011).

In vector magnetometers, there are slight errors in the orthogonality of the measurement elements. To compensate for these errors, the sensor is calibrated. Calibration for orthogonality requires a large number of measurements to be gathered with the magnetometer in a constant magnetic field at as many different attitudes as possible. The data can then be taken and transformed from an ellipsoid to the sphere that the measurements should represent. This computed correction is then applied to future measurements after temperature compensation (Merayo et al., 2000).

In the low-cost sensor chosen, the internal errors in the sensing element are significantly lower than the errors due to noise in the readout electronics. To determine the nature of this noise, a number of measurements were taken over time and analysed. Kalman filtering assumes the measurement noise is Gaussian white noise (Welch and Bishop, 2006), so tests were run to confirm these assumptions. The Fourier transform of the collected data, in Figure 2-3 (a) was taken to show the noise was white. The spectral analysis shows the noise is white as it is independent of frequency. The only peak in the frequency spectrum is at 60 Hz, which is consistent with the frequency of power wiring in the building where the measurements were taken, and not caused by the sensor noise. Figure 2-3 (b) shows the distribution of the measurements taken plotted against the normal distribution. Testing the sample for normality showed the normal function to be a good approximation of the noise in the data. The measurements and their frequency spectrum confirm that the noise is roughly white and Gaussian in nature.

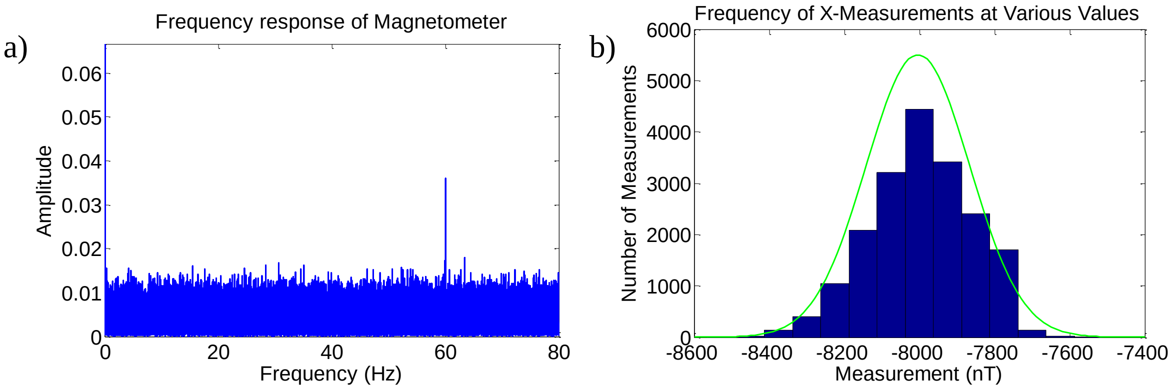

Figure 2-3 Noise profile of x-axis magnetometer measurements. a) Frequency response of measurements. b) Distribution of measurements in relation to the normal probability function.

2.1.4 Effects of Magnetorquers on Satellites

Magnetorquers are electromagnets used to generate torques on the satellite as they interact with the Earth's magnetic field. There are two types used on nanosatellites, magnetic torque rods and large air core solenoids. Torque rods consist of many turns of wire around long cylinders of high permeability core material, which amplifies the magnetic field generated by the coil, producing a stronger magnetic dipole. Air core solenoids are generally the size of an entire side of the satellite and use the increased area to generate increased magnetic fields. Air core solenoids are generally weaker than cored rods. Both types generate a magnetic dipole based on the voltage applied to them. The magnetic dipole causes a torque on the satellite, which can be calculated from:

$$\vec{\tau}_{MAG} = \vec{d} \times \vec{b} \tag{2-1}$$

where $\vec{b}$ is the external magnetic field, $\vec{d}$ is the magnetic dipole moment of the magnetorquers, and $\vec{\tau}_{MAG}$ is the torque produced by the magnetorquers.

Magnetorquers are the most significant source of magnetic field on a nanosatellite. They can generate a magnetic field strong enough to ruin measurements from an onboard magnetometer. There are several ways to mitigate the effects magnetorquers have on measurements. The first is to subtract the field generated by the magnetorquers from the measurements. Subtracting the expected field can add an extra source of bias and noise if the magnetorquers are producing different field strengths than expected. It also requires the magnetometer to accommodate much higher field strengths than it would normally need to measure. The second method is to only take measurements when the magnetorquers are inactive. This process reduces the time that the magnetorquers and magnetometer are running, as they must not operate simultaneously, which provides an improvement in power consumption, but a degradation of control. For cored magnetorquers, residual magnetic fields may remain and add small biases or soft ferromagnetic errors based on the residual magnetization left in the magnetorquers. Air core magnetorquers have the advantage of not needing to demagnetize the core for this method to work. The third method is to mount the magnetometer on a boom to allow measurements to be taken with minimal interference from any satellite-generated magnetic fields. An improvement on this method involves mounting two magnetometers at varying distances on the boom and removing the largest of the satellite sources of magnetic fields in the calculations, as they are proportional to $\frac{1}{r^3}$, where r is the distance between the magnetometer and the centre of the assumed dipole (Acuña, 2002).

2.2 Models of the Earth's Magnetic Field

In modelling the magnetic field around the Earth, there are two fields that need to be examined, the field generated by the Earth's internal geodynamo and crust, and the fields generated externally to the Earth. The internal field is modelled frequently to a high degree of accuracy by several organizations. The World Magnetic Model (WMM), IGRF, and Enhanced Magnetic Model (EMM), are produced every 5 years, with predictive secular variations (Nair, 2014b). The High Definition Geomagnetic Model (HDGM) is produced every year with secular variations (Nair, 2014a). The external field is produced by the Sun and the motion of the charged particles it emits. A model for the external field was created by Olson and Pfitzer (1974), among others at various times.

Mathematical models approximate the Earth's internal magnetic field using a spherical harmonic expansion. As the region of interest is several hundred kilometres above the Earth's surface, the higher order spherical terms have less and less of an effect. The added accuracy of the EMM and HDGM, both of degree and order 720, gives too little gain, given the low resolution of the magnetometer, at too high a computational cost. The WMM has similar accuracy to the sensors used, as it is only required to be within 280 nT of the true value, with more emphasis placed on the accuracy of declination values for navigation than on field strength (Maus et al., 2009). The IGRF, being accurate to 0.1%, or approximately 30 nT, shortly after publication, is more accurate than the WMM (Kianfar et al., 2011). For this reason, the IGRF is used, despite the added computational complexity of 13th order over 12th order spherical harmonics.

To model the external field, several methods exist. However, only one model, the Olson-Pfitzer model, is available for STK. The Olson-Pfitzer model uses power series and exponential expansions to model the magnetosphere on quiet days. It accounts for the three main sources of the external magnetic field: the magnetopause, neutral sheet, and ring current (Community Coordinated Modeling Center, 2011), and assumes a fixed direction for the main field, and ring currents (Pfitzer and Olson, 2011).

For simulation purposes, the internal and external fields are modelled using the IGRF and Olson-Pfizer models. The EKF constructed only uses the IGRF, as this is normally the only model used on nanosatellites. The Olson-Pfizer model is not normally implemented on nanosatellites due to the computational burden and little gain for low orbiting satellites (Atmospheric and Environmental Research Inc., 2014).

2.2.1 International Geomagnetic Reference Field (IGRF)

In order to model the geomagnetic field, it is generally expressed as a spherical harmonic expansion of a scalar potential field, s (Merrill and McElhinny, 1982). To obtain the magnetic field vector, $\vec{b}$, the gradient of s is taken, which leads to equations for $\vec{b}$ and s of the form (Malin, 1987):

$$\vec{b}(r,\theta,\phi) = \nabla s(r,\theta,\phi) \tag{2-2}$$

$$\begin{aligned} s(r,\theta,\phi) = r_E \sum_{j=1}^{\infty}\sum_{i=0}^{j} \Bigg\{ &\left[g_{j,i}\cos i\phi + h_{j,i}\sin i\phi\right]\left(\frac{r_E}{r}\right)^{j+1} \\ &+ \left[e_{j,i}\cos i\phi + f_{j,i}\sin i\phi\right]\left(\frac{r}{r_E}\right)^{j}\Bigg\} p_j^i(\cos\theta) \end{aligned} \tag{2-3}$$

where r_E is the radius of the Earth; $g_{j,i}$, $h_{j,i}$, $e_{j,i}$, and $f_{j,i}$ are the spherical harmonic coefficients; j is the degree of the model; i is the order of the model; r is the distance from the centre of the Earth; θ is the colatitude; ϕ is the longitude; and $p_j^i(\cos\theta)$ are the associated Legendre functions. The terms in $\left(\frac{r_E}{r}\right)^{j+1}$ are from sources inside the Earth, and terms in $\left(\frac{r}{r_E}\right)^{j}$ are from sources external to the Earth (Malin, 1987). In order to increase the ability to compare different models, Schmidt quasi-normalized functions are used instead of the normal associated Legendre functions (Winch et al., 2004). Schmidt quasi-normalized functions take the form (Winch et al., 2004):

$$p_j(u) = \frac{1}{2^j j!}\left(\frac{d}{du}\right)^j (u^2 - 1)^j \tag{2-4}$$

$$p_j^i(u) = \sqrt{2\frac{(j-i)!}{(j+i)!}}(1-u^2)^{i/2}\left(\frac{d}{du}\right)^i p_j(u) \tag{2-5}$$

The magnetic field vector can be found by taking the gradient of s, neglecting the terms in $\left(\frac{r}{r_E}\right)^j$ as only internal fields are modelled in the IGRF. In ECEF coordinates, Montenbruck and Gill (2005) show the gradient of a spherical harmonic potential function to be:

$$\vec{b}_x = \sum_{j=1}^{k}\sum_{i=0}^{j}\vec{b}_{x,j,i} \tag{2-6}$$

$$\vec{b}_y = \sum_{j=1}^{k}\sum_{i=0}^{j}\vec{b}_{y,j,i} \tag{2-7}$$

$$\vec{b}_z = \sum_{j=1}^{k}\sum_{i=0}^{j}\vec{b}_{z,j,i} \tag{2-8}$$

$$\vec{b}_{z,j,i} = (j-i+1)\left[-g_{j,i}v_{j+1,i} - h_{j,i}w_{j+1,i}\right] \tag{2-9}$$

When $i = 0$,

$$\vec{b}_{x,j,0} = \left[-g_{j,0}v_{j+1,1}\right] \tag{2-10}$$

$$\vec{b}_{y,j,0} = \left[-g_{j,0}w_{j+1,1}\right] \tag{2-11}$$

When $i > 0$,

$$\vec{b}_{x,j,i} = \frac{1}{2}\left[\left(-g_{j,i}v_{j+1,i+1} - h_{j,i}w_{j+1,i+1}\right) + \frac{(j-i+2)!}{(j-i)!}\left(g_{j,i}v_{j+1,i-1} + h_{j,i}w_{j+1,i-1}\right)\right] \tag{2-12}$$

$$\vec{b}_{y,j,i} = \frac{1}{2}\left[\left(-g_{j,i}w_{j+1,i+1} + h_{j,i}v_{j+1,i+1}\right) + \frac{(j-i+2)!}{(j-i)!}\left(-g_{j,i}w_{j+1,i-1} + h_{j,i}v_{j+1,i-1}\right)\right] \tag{2-13}$$

where,

$$v_{j,i} = \left(\frac{r_E}{r}\right)^{j+1} p_{j,i}(\cos\theta)\cos i\phi \tag{2-14}$$

$$w_{j,i} = \left(\frac{r_E}{r}\right)^{j+1} p_{j,i}(\cos\theta)\sin i\phi \tag{2-15}$$

Equations (2-9) through (2-15) use the normal definition of associated Legendre functions,

$$p_j(u) = \frac{1}{2^j j!}\frac{d^j}{du^j}(u^2-1)^j \tag{2-16}$$

$$p_{j,i}(u) = (1-u^2)^{i/2}\frac{d^i}{du^i}p_j(u) \tag{2-17}$$

Converting equations (2-9) through (2-15) to use Schmidt quasi-normalized associated Legendre functions, where δ_i^0 is the Kronecker delta,

$$\delta_i^0 = \begin{cases} 1 & , i = 0 \\ 0 & , i \neq 0 \end{cases} \tag{2-18}$$

$$p_j^i = \sqrt{\frac{(2-\delta_i^0)(j-i)!}{(j+i)!}}\,p_{j,i} \tag{2-19}$$

$$v_j^i = \sqrt{\frac{(2-\delta_i^0)(j-i)!}{(j+i)!}}\,v_{j,i} \tag{2-20}$$

$$w_j^i = \sqrt{\frac{\left(2-\delta_i^0\right)(j-i)!}{(j+i)!}}\,w_{j,i} \tag{2-21}$$

$$g_j^i = \sqrt{\frac{(j+i)!}{\left(2-\delta_i^0\right)(j-i)!}}\,g_{j,i} \tag{2-22}$$

$$h_j^i = \sqrt{\frac{(j+i)!}{\left(2-\delta_i^0\right)(j-i)!}}\,h_{j,i} \tag{2-23}$$

and the magnetic field vector becomes,

$$\begin{aligned}\vec{b}_{x,j,i} = &\left[\frac{\sqrt{2}\left[-g_j^i v_{j+1}^{i+1} - \left(1-\delta_i^0\right) h_j^i w_{j+1}^{i+1}\right]}{\sqrt{\left(2-\delta_i^0\right)^3 (j+i+2)(j+i+1)}}\right. \\ &\left. + \left(1-\delta_i^0\right)\sqrt{\frac{\left(2-\delta_i^0\right)(j-i+2)!^3}{(j-i)!^3}}\left(g_j^i v_{j+1}^{i-1} + h_j^i w_{j+1}^{i-1}\right)\right]\end{aligned} \tag{2-24}$$

$$\begin{aligned}\vec{b}_{y,i,j} = &\left[\frac{\sqrt{2}\left[-g_j^i w_{j+1}^{i+1} - \left(1-\delta_i^0\right) h_j^i v_{j+1}^{i+1}\right]}{\sqrt{\left(2-\delta_i^0\right)^3 (j+i+2)(j+i+1)}}\right. \\ &\left. + \left(1-\delta_i^0\right)\sqrt{\frac{\left(2-\delta_i^0\right)(j-i+2)!^3}{(j-i)!^3}}\left(g_j^i w_{j+1}^{i-1} + h_j^i v_{j+1}^{i-1}\right)\right]\end{aligned} \tag{2-25}$$

$$\vec{b}_{z,j,i} = \sqrt{\frac{(j-i+1)^3}{(j+i+1)}}\left[-g_j^i v_{j+1}^i - h_j^i w_{j+1}^i\right] \tag{2-26}$$

with the partial derivatives of these equations, which are necessary for the EKF, included in Appendix B. The spherical harmonic coefficients g_j^i and h_j^i are published every 5 years by the International Association of Geomagnetism and Aeronomy and are available for download on their website (Nair, 2014b).

3 Attitude Determination

To provide a baseline for magnetometer-only attitude determination on a nanosatellite, the EKF for attitude determination is presented. First, the representation of the attitude is described. Second, the dynamics governing this representation are presented. Third, the particular formulation of the EKF used is presented. And lastly, a baseline is developed by examining the response of the attitude determination EKF under simulated conditions.

3.1 Attitude Representation using Quaternions and Angular Rates

One method for representing an object's attitude is by using a quaternion. This representation is based on the ability to obtain any attitude by rotating an object around an arbitrary axis by an arbitrary amount. The advantages of using quaternions are that they do not contain singularities, and that they can be expressed using three independent terms. For these reasons, they are employed in the attitude determination EKF. Quaternions store the unit vector of the rotation axis, $\hat{u}$, and the right-handed rotation about that axis, θ, in four variables such that their norm is 1 (Abdelrahman and Park, 2011), as shown in equation (3-2), to prevent unwanted scaling:

$$q = \begin{bmatrix} \vec{q} \\ q_4 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \vec{\boldsymbol{u}}_x \sin(\theta/2) \\ \vec{\boldsymbol{u}}_y \sin(\theta/2) \\ \vec{\boldsymbol{u}}_z \sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix} \tag{3-1}$$

$$\sqrt{{q_1}^2 + {q_2}^2 + {q_3}^2 + {q_4}^2} = 1 \tag{3-2}$$

This constraint causes the quaternion to have only 3 independent variables. It is common practice to take the scalar part, q_4, as the dependant term, and the vector part, $\vec{q}$, as the independent terms. While quaternions do not have singularities, they are not unique. A quaternion and its negative generate the same rotation.

To rotate a vector, $\vec{u}$, from one reference frame, C, to another, D, quaternion multiplication, $\otimes$, is used, where the scalar part of $\vec{b}$ is 0, and $q^{D/C}$ is the quaternion which defines reference frame C as a rotation from reference frame D.

$$\begin{bmatrix}\vec{b}^D\\ 0\end{bmatrix} = q^{D/C} \otimes \begin{bmatrix}\vec{b}^C\\ 0\end{bmatrix} \otimes q^{C/D} \tag{3-3}$$

To combine quaternions, quaternion multiplication is also used:

$$q^{E/C} = q^{E/D} \otimes q^{D/C} \tag{3-4}$$

where quaternion multiplication is defined as:

$$q \otimes q' = \begin{bmatrix} q_1 q_4' + q_2 q_3' - q_3 q_2' + q_4 q_1' \\ -q_1 q_3' + q_2 q_4' + q_3 q_1' + q_4 q_2' \\ q_1 q_2' - q_2 q_1' + q_3 q_4' + q_4 q_3' \\ -q_1 q_1' - q_2 q_2' - q_3 q_3' + q_4 q_4' \end{bmatrix} \tag{3-5}$$

Taking equations (3-5) and (3-3), the equivalent rotation matrix can be obtained:

$$\mathcal{R}^{D/C}\left(q^{D/C}\right) = \begin{bmatrix} 1 - 2{q_2}^2 - 2{q_3}^2 & 2(q_1 q_2 + q_3 q_4) & 2(q_1 q_3 - q_2 q_4) \\ 2(q_1 q_2 - q_3 q_4) & 1 - 2{q_1}^2 - 2{q_3}^2 & 2(q_2 q_3 + q_1 q_4) \\ 2(q_1 q_3 + q_2 q_4) & 2(q_2 q_3 - q_1 q_4) & 1 - 2{q_1}^2 - 2{q_2}^2 \end{bmatrix} \tag{3-6}$$

$$\vec{b}^D = \mathcal{R}^{D/C} \vec{b}^C \tag{3-7}$$

The quaternion representing the reverse rotation is:

$$q^{C/D} = {q^{D/C}}^* = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ -q_4 \end{bmatrix} \tag{3-8}$$

To represent the rate of change of the attitude, angular rates can be used. The angular rates of reference frame C with respect to reference frame D is $\vec{\omega}^{D}/_{C}$. To combine angular rates, simple addition is used.

$$\vec{\omega}^{E}/_{C} = \vec{\omega}^{E}/_{D} + \vec{\omega}^{D}/_{C} \tag{3-9}$$

The reverse angular rates are:

$$\vec{\omega}^{C}/_{D} = -\vec{\omega}^{D}/_{C} \tag{3-10}$$

Quaternion kinematics describe the change in a quaternion based on the angular rates of the associated reference frame according to:

$$\dot{q} = \frac{1}{2}\,\Omega(\vec{\omega})q = \frac{1}{2}\Xi(q)\vec{\omega} \tag{3-11}$$

where $\Omega(\vec{\omega})$ and $\Xi(q)$ are:

$$\Omega(\vec{\omega}) = \begin{bmatrix} 0 & \vec{\omega}_z & -\vec{\omega}_y & \vec{\omega}_x \\ -\vec{\omega}_z & 0 & \vec{\omega}_x & \vec{\omega}_y \\ \vec{\omega}_y & -\vec{\omega}_x & 0 & \vec{\omega}_z \\ -\vec{\omega}_x & -\vec{\omega}_y & -\vec{\omega}_z & 0 \end{bmatrix} \tag{3-12}$$

$$\Xi(q) = \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \tag{3-13}$$

The time rate of change in angular rates based on the torque, $\vec{\tau}$, are found according to the Newton-Euler equation for a rigid body.

$$\dot{\vec{\omega}} = I^{-1}[\vec{\tau} - \vec{\omega} \times (I\vec{\omega})] \tag{3-14}$$

3.2 Disturbance Torques Included in the Simulations

There are a number of torques that affect the rate of change of the angular rates, and thus the attitude, of satellites: magnetic torques, gravity gradient torques, aerodynamic torques, and solar radiation pressure torques. In order to determine the steady state performance of the attitude determination algorithm, it is necessary to examine the response of the algorithm under controlled conditions. To prevent the satellite from tumbling in the simulations, it is assumed that the attitude control algorithm is countering the majority of the disturbance torques. Only the gravity gradient torque is included to highlight the differences between placing the magnetometer on a boom and placing it inside the satellite.

3.2.1 Gravity Gradient Disturbance Torques

The gravitational force due to the Earth varies with distance, “thus the gravitational force on one part of [a] spacecraft is different from that on another, and this difference results in a net torque” (Pisacane and Moore, 1994), called gravity gradient torque. Gravity gradient torque causes the satellite to rotate toward an equilibrium attitude where the minor axis points toward the Earth. The torques generated, τ_{GG}, are calculated according to:

$$\tau_{GG} = \frac{3\kappa}{|\vec{r}^S|^3}[\vec{r}^S \times (I \cdot \vec{r}^S)] \tag{3-15}$$

where $\vec{r}^S$ is the position of the centre of the Earth in the satellite body frame, I is the moment of inertia of the satellite, and κ is the Earth’s gravitational constant (Wertz, 1978).

3.3 Implemented Extended Kalman Filter for Attitude Determination

For estimating the attitude, a subset of the EKF, the Multiplicative Extended Kalman Filter, MEKF, as described by Markley (2003), is used. The state vector is comprised of independent variables that fully describe the system being estimated, namely the error quaternion and angular rates. As the quaternion is needed to describe the attitude, it is kept as a reference quaternion, q_{REF}. The vector part of the error quaternion, $\delta\vec{q}$, is used, with the scalar part, δq_4, taken as dependent, leading to the state vector, $\vec{x}$:

$$\vec{x} = \begin{bmatrix} \delta\vec{q} \\ \vec{\omega} \end{bmatrix} \tag{3-16}$$

The error quaternion, δq, is used to update the reference quaternion. The error quaternion is reset to 0 by updating the reference quaternion before and after the measurement update. The reset is done using quaternion multiplication shown in equation (3-4). The dependent term in the error quaternion is found using equation (3-2) and taking the positive root.

For the purpose of the MEKF, the reference quaternion defines the estimated satellite reference frame in terms of a rotation from the inertial frame, the error quaternion defines the true satellite reference frame in terms of a rotation from the estimated satellite reference frame, and the angular rate term is the rotation of the satellite frame with respect to the inertial frame.

3.3.1 Dynamic Update of the Attitude State Vector

The a priori state vector, $\vec{x}_t^-$, is found by integrating equations (3-11) and (3-14). For the integration of both equations, fourth order Runge-Kutta numerical integration methods (Süli and Mayers, 2003) are employed.

$$\vec{x}_t^- = g(\hat{\vec{x}}_{t-1}) = \hat{\vec{x}}_{t-1} + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{3-17}$$

where the integration terms k_1 to k_4 are found from the function describing the time rate of change of the state vector based on the state vector itself, $g(\vec{x})$, at different points along the curve between the current time and the time integrated to.

$$k_1 = \Delta t g(\vec{x}) \tag{3-18}$$

$$k_2 = \Delta t g\left(\vec{x} + \frac{1}{2}k_1\right) \tag{3-19}$$

$$k_3 = \Delta t g\left(\vec{x} + \frac{1}{2}k_2\right) \tag{3-20}$$

$$k_4 = \Delta t g(\vec{x} + k_3) \tag{3-21}$$

with the rate of change of the state vector, $\dot{\vec{x}}$, taken from equations (3-11) and (3-14), and assuming the external torques are negligible:

$$\dot{\vec{x}} = g(\vec{x}) = \begin{bmatrix} \frac{1}{2}\,\Omega(\vec{\omega})q \\ I^{-1}[-\vec{\omega} \times (I\vec{\omega})] \end{bmatrix} \tag{3-22}$$

3.3.2 Measurement Update of the Attitude State Vector

The measurement update is performed in two steps. First, the measurement is estimated from a priori information. Then, the state is updated to reflect the actual measurement. The a priori measurement is calculated using the measurement model from equation (1-3), using the reference quaternion to rotate the magnetic field vector from the IGRF into the body frame (Abdelrahman and Park, 2011). The measurement model is:

$$\vec{z}^- = h(\vec{x}^-) = q_{REF}^- \otimes \vec{b}^I \otimes q_{REF}^- = \mathcal{R}_{REF}^- \vec{b}^I \tag{3-23}$$

where $\vec{b}^I$ is the magnetic field in the inertial reference frame, calculated from the IGRF. The estimated state vector is found according to equation (1-5). The reference quaternion is then updated using quaternion multiplication, and the error quaternion in the state vector is set to 0:

$$\hat{q}_{REF,t} = q_{REF,t}^- \otimes \delta\hat{q}_t \tag{3-24}$$

$$\delta\hat{q}_t = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{3-25}$$

In order to evaluate the attitude determination MEKF and set up a baseline for the performance of the MEKF, the error in the estimated attitude is needed. The error in the estimated attitude is obtained by rearranging equations (3-4) and (3-8), and (3-9) and (3-10):

$$\delta q^{S}/_{\hat{s}} = q^{I}/_{s}{}^{*} \otimes q^{I}/_{\hat{s}} = q^{I}/_{s}{}^{*} \otimes \hat{q}_{REF} \tag{3-26}$$

$$\delta \vec{\omega}^{S}/_{\hat{s}} = -\vec{\omega}^{I}/_{s} + \vec{\omega}^{I}/_{\hat{s}} = \widehat{\omega} - \vec{\omega}^{I}/_{s} \tag{3-27}$$

where $q^{I}/_{s}$ represents the true attitude quaternion, and $\vec{\omega}^{I}/_{s}$ represents the true angular rates of the satellite.

3.4 Simulation Studies Used in Evaluating the Attitude Determination EKF

To demonstrate the performance of the attitude determination filter, several simulations were run. The simulations used typical nanosatellite orbit parameters. A nearly circular, Sun synchronous, noon-midnight, orbit at an apogee height of 400 km, shown in Table 3-1, was used as it is typical of many of the surveyed nanosatellite missions. More detail is presented on the variety and distribution of typical nanosatellite orbits in section 5.2. The true attitude was initially set as nadir pointing, as many nanosatellites perform Earth observation. The attitude was perturbed from this orientation by the gravity gradient torque, showing the effects that a large boom can have on the attitude determination. The specific initial attitude parameters are shown in Table 3-2. Initial attitude parameters were chosen to show how the MEKF would perform in the event that it took over from a previous system, case ‘a’, and with no prior knowledge of its attitude, cases ‘b’ and ‘c’. Case ‘a’ started with an initial error of 11°, slightly more than what was expected of the attitude determination MEKF. Cases ‘b’ and ‘c’ were started with over 120° of error, showing the ability of the filter to converge under adverse circumstances.

Table 3-1 Initial orbital parameters used for attitude determination simulations.

Semi-major Axis (Earth radii)	1.0627142
Eccentricity	0.0015
Inclination (rad)	1.6935733
Argument of Perigee (rad)	0
RAAN (rad)	1.7719979
Mean Anomaly (rad)	0

3.4.1 Initialization of the State Vector for the Attitude Determination Simulations

Using the magnetic field vector only allows 2 axes of information, so not enough information is available from the chosen sensor to properly initialize the system. It was therefore necessary to make the system robust enough to handle a lack of proper initialization. To test the MEKF two different initialization cases were used: one with low initial error, case ‘a’, to show steady state performance characteristics in the event that the system takes over from another system; and the other with large initial error, case ‘b’, to show filter convergence. A third case, case ‘c’, showing the effect a long boom can have was also examined by changing the moment of inertia matrix to that of a nanosatellite with a long boom. The initial conditions for all three cases are shown in Table 3-2.

Table 3-2 Initial attitude conditions used for attitude determination simulations.

	True Values	Case a (low initial error)	Case b (high initial error)	Case c (extended boom)
$\vec{\omega}_x$ (rad/s)	0	0	0	0
$\vec{\omega}_y$ (rad/s)	-1.02 x10^{-3}	-1x10^{-3}	0	0
$\vec{\omega}_z$ (rad/s)	2.30x10^{-8}	2x10^{-8}	0	0
q_1	-0.695	-0.7	0.5	0.5
q_2	-0.125	-0.1	0.5	0.5
q_3	-0.707	-0.7	0.5	0.5
q_4	0.0148	0.1	0.5	0.5
I_{xx} (kg m^2)	–	0.00283	0.00283	0.0423
I_{yy} (kg m^2)	–	0.00247	0.00247	0.0422
I_{zz} (kg m^2)	–	0.00314	0.00314	0.00283

For case ‘a’, the initial reference quaternion had an error of 11°. The initial angular rates were obtained by rounding the true values to 1 significant figure. To accommodate the large initial error of case ‘b’, a large covariance matrix was chosen with all of the state variables assumed to be independent. The chosen initial reference quaternion had an error of 126°. Even though this error is quite high, care needed to be taken in the selection of quaternion covariances to ensure the estimated error quaternion did not significantly exceed its normalization. For this reason, the covariances were chosen to be 0.1 for all quaternion terms, which equates to a standard deviation of 0.316. The chosen covariances make the filter unlikely to calculate an error quaternion that is high enough in all three axes as to no longer be able to satisfy the constraint equation (3-2). The initial angular rates were set to 0 radians per second. This value was chosen as, with no other knowledge of the rotation of the satellite, it was central to all possible values that could be encountered. The covariance of this initial estimate was set to 0.0001 as the satellite was already under control and rotating slowly. Case ‘c’ used a large initial error to show convergence under more heavily perturbed conditions, and therefore used the same initial conditions as case ‘b’.

3.4.2 Simulation Results for the Attitude Determination EKF

Based on the results from Psiaki (1990) and Abdelrahman (2011), the MEKF should have enough information to converge to the correct attitude within one orbit to an accuracy of approximately 5°. Figure 3-1 shows the initial convergence period of the three cases. Case ‘a’, starting with little initial error, converged almost immediately. The filter converged quickly in both cases ‘b’ and ‘c’, taking only 0.5 orbits and 0.6 orbits to reach a steady state, respectively. Even though both cases ‘b’ and ‘c’ have the same initial conditions, case ‘c’ took longer to converge due to the larger disturbance torques it included. This quick convergence was well within what was expected. Attitude MEKF convergence faster than 0.6 orbits will have little impact on the ability of a nanosatellite to perform its mission, as it typically only needs to be performed once, taking less than one hour of time in a mission lasting months or years.

As cases ‘a’ and ‘b’ used the same attitude and orbit parameters, they showed nearly identical results. The similar time response between cases ‘a’ and ‘b’ can be attributed to the unmodelled disturbance torques and initial orbit and attitude conditions, which were the same for both cases. Case ‘c’ had similar, though significantly larger, peaks in the attitude error as the gravity gradient torque was still unmodelled, and much larger in this case.

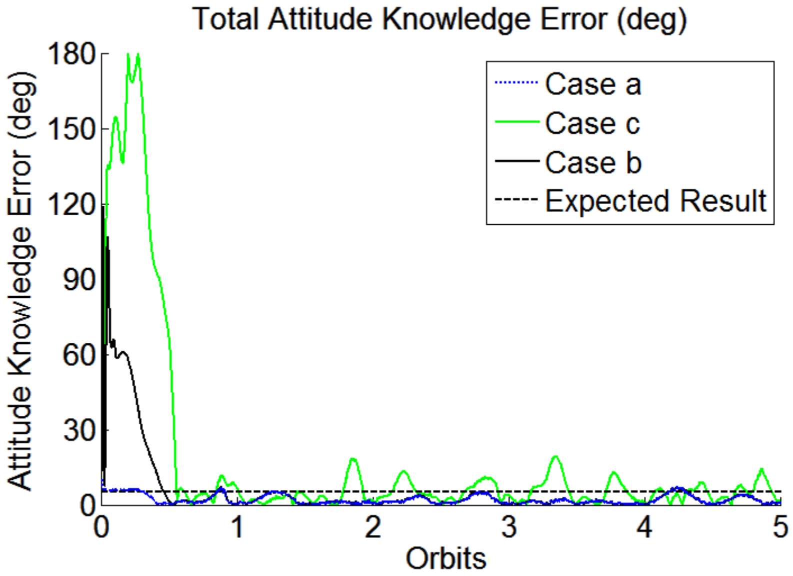

Figure 3-1 Attitude MEKF response for all three test cases.

The steady state accuracy of case ‘a’ was expected to be the best, as less error was included in initialization. Case ‘b’ was expected to have similar steady state accuracy as case ‘a’, as the disturbance environment and orbit were the same. Case ‘c’ was expected to have larger steady state errors as it included greatly increased disturbance torques due to the inclusion of a large boom in the moment of inertia matrix. In each case where the disturbance torques were small, the accuracy was close to the 5° that was expected. Table 3-3 shows the steady state accuracy of the attitude determination MEKF in each case through the 95th percentile of the overall pointing knowledge. This accuracy serves as a baseline to which subsequent results can be compared.

Table 3-3 95th percentile of error in overall steady state attitude estimate for all three test cases.

	Case a	**Case b**	**Case c**
Attitude knowledge error (°)	5.55	5.55	13.0

Even though the filter converged as expected, the covariance estimate did not reflect the accuracy of the estimated parameters during the convergence period. The covariance converged in less than one minute, shown in Figure 3-2 (a), while it took much longer, almost one hour, for the state estimate to converge, shown in Figure 3-2 (b). The covariance matrix was updated to reflect the agreement between the actual measurement and the estimated measurement as part of the MEKF. Given that this agreement was independent of rotation in one axis, it could be quite high without the estimated attitude accurately reflecting the true attitude, which caused the covariances to converge much more quickly than the estimated attitude. This discrepancy made it difficult to tell when the filter had converged without the aid of a reference solution. It may be possible to reduce the discrepancy through tuning, or by using additional sensors. Identification of the convergence period may be possible by using a smoothing estimator to estimate previous epochs using current measurements and check their agreement with the results of the MEKF. Smoothing would, however, be very computationally intensive for the short convergence period. Initializing the MEKF with an estimate that is already accurate to approximately 5°, by utilizing Sun sensors or other sensors, can also be used to eliminate the convergence time, as with case ‘a’.

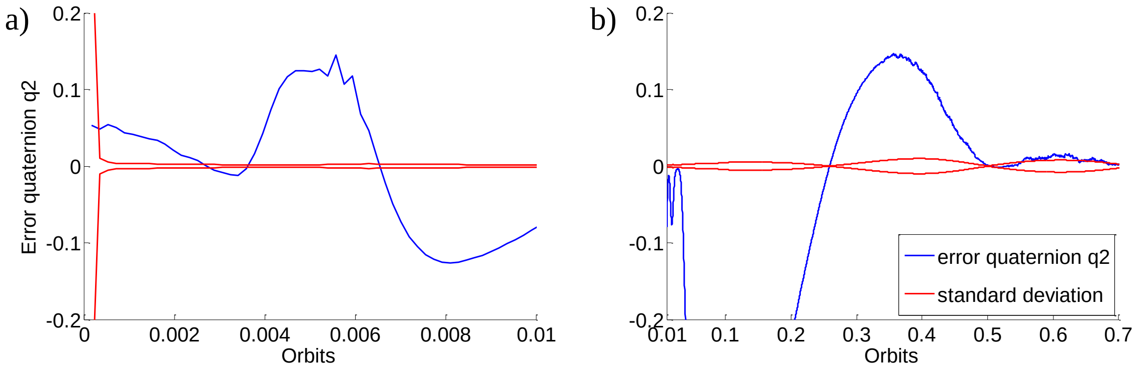

Figure 3-2 Representative covariance convergence from case b, showing discrepancy between a) convergence of estimated standard deviation, and b) convergence of error quaternion term q2.

3.5 Summary of the Attitude Determination Extended Kalman Filter

An attitude determination algorithm based on an MEKF was developed to estimate the error in a reference quaternion and the angular rates of a nanosatellite. The reference quaternion and angular rates were propagated using fourth order Runge-Kutta numerical integration of the quaternion kinematics and Newton-Euler equations. The MEKF accomplished the estimation by comparing three-axis magnetic field measurements to the IGRF. Three test cases have shown that the MEKF created is able to handle the attitude determination needs of nanosatellites with internal magnetometers as well as magnetometers on booms. The MEKF took only 50% of the expected 1 orbit to converge. The tests have set the baseline for attitude determination to be used for comparison with combined orbit and attitude determination at a convergence time of 0.5 orbits and an accuracy of 5.5°. The MEKF had difficulty identifying convergence, so it should only be used as a backup to other systems, where additional sensors can aid in initializing the MEKF or determining convergence. Improvements may be possible with further tuning, and the inclusion of disturbance torques in the dynamic model.

4 Orbit Determination

To provide a baseline for magnetometer-only orbit determination on a nanosatellite, the EKF for orbit determination is presented. First, the representation of the orbit is described. Second, the dynamics governing this representation are presented. Third, the particular formulation of the EKF used is presented. And lastly, a baseline is developed by examining the response of the orbit determination EKF under simulated conditions.

4.1 Representations of Orbits Used in the Orbit Determination EKF

In order to fully define a satellite's orbit, as a solution to the special case of the two body problem where the mass of the satellite is negligible, six terms are required. There are several representations used to express these six terms. Commonly, the six Cartesian terms of position and velocity, or the classical orbital elements: semi-major axis, eccentricity, inclination, argument of perigee, right ascension of the ascending node, and mean anomaly are used.

4.1.1 Position and Velocity Representation of Orbits

As only forces act on a satellite, its orbit can be fully defined by its position and velocity. An advantage of using position and velocity to define the orbit of a satellite is that many models require Cartesian position as an input. Also, many disturbance accelerations are expressed in a Cartesian system. Lastly, for analysing the payload data, it is easier in many cases to use Cartesian positions. The disadvantage of using a Cartesian system is that the non-linear equations of motion become more difficult to propagate.

Cartesian position and velocity can be expressed in several reference frames. Many Earth-based models use an Earth-Centred, Earth-Fixed (ECEF) coordinate system, shown in Figure

4-1. In the ECEF system, the origin is at the centre of the Earth and the axes are aligned such that the Z-axis points north along the axis of rotation of the Earth, the X-axis points along the prime (or Greenwich) meridian at the equator, and the Y-axis points along the equator, completing the right-handed coordinate system. The ECEF frame is often called the International Terrestrial Reference Frame, or ITRF.

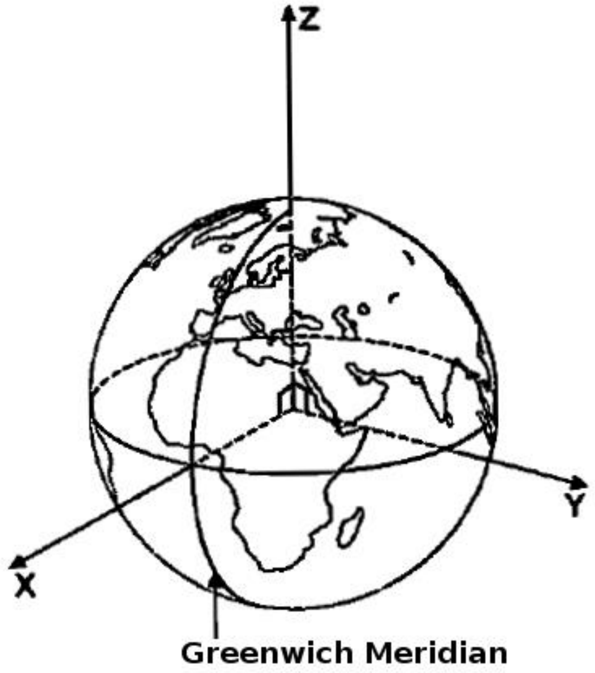

Figure 4-1 Definition of the Earth Centred Earth Fixed (ECEF) coordinate system (Larson and Wertz, 1999).

The Earth Centred Inertial (ECI) coordinate system, often called the J2000 frame, is used frequently for astronomical models, such as Sun and Moon ephemerides, as well as for satellites. The ECI frame’s axes are fixed with respect to inertial space and do not change over time, which makes it especially convenient for representations of satellite attitude. This frame has the Z-axis pointing north along the mean axis of rotation of the Earth, and the X-axis pointing toward the mean vernal equinox as they were at 12:00 on January 1, 2000.

The True Equator Mean Equinox (TEME) frame is an intermediate frame often employed on satellites using the SGP4 propagator for orbit determination. It is the frame in which the SGP4 propagator gives the calculated position and velocity from NORAD TLEs. This frame has the Z-axis pointing north along the axis of rotation of the Earth, and the X-axis pointing toward the mean equinox.

Position and velocity can be converted from one frame to another by employing a rotation matrix. Generally, the conversion from one frame to the next can be expressed as:

$$\vec{r}^{A} = \mathcal{R}^{A/B}\vec{r}^{B} \tag{4-1}$$

$$\dot{\vec{r}}^{A} = \mathcal{R}^{A/B}\dot{\vec{r}}^{B} + \vec{\omega}^{A/B} \times \vec{r}^{B} \tag{4-2}$$

where $\mathcal{R}^{A/B}$ is the rotation matrix from frame B to frame A, and $\vec{\omega}^{A/B}$ is the angular rates of frame B with respect to frame A.

In particular, the conversion from the ECEF frame to the TEME frame and then to the ECI frame, with intermediate pseudo-Earth fixed (PEF), true of date (TOD), and mean of date (MOD) frames, is:

$$\vec{r}^{ECEF} = \mathcal{R}_{PM}^{ECEF/PEF}\mathcal{R}_{GMST}^{PEF/TEME}\mathcal{R}_{EQE}^{TEME/TOD}\mathcal{R}_{NUT}^{TOD/MOD}\mathcal{R}_{PRE}^{MOD/ECI}\vec{r}^{ECI} \tag{4-3}$$

where the subscripts indicate the rotation matrices that account for polar motion (PM), sidereal time (GMST), equation of the equinoxes (EQE), precession (PRE), and nutation (NUT). The polar motion is small, so it is often neglected for nanosatellites without causing a significant loss in accuracy. The precession and nutation change slowly, so they require infrequent recalculation. Given that the inverse of a rotation matrix is the same as its transpose, it becomes easy to rearrange equation (4-3) to change between any of these frames.

For the purposes of orbit determination, the polar motion is neglected as it causes a loss in accuracy on the order of 0.0001% in the position. The equation of the equinoxes, precession, and nutation, which account for a combined loss in accuracy on the order of 0.1%, are neglected in the rotation of the magnetic field vector to the inertial frame as the loss of accuracy is less than the accuracy of the sensors. Those rotations are neglected in order to reduce the computational burden of the algorithm as has been done previously in Gregory (2001). For generating the reference solution, all rotation matrices are used.

4.1.2 Orbital Element Representation of Orbits – SGP4 Representation

Keplerian orbital elements, shown in Figure 4-2, describe the solution to the two body problem in graphical terms. They describe the elliptical path by its semi-major axis, α, and eccentricity, ε. The attitude of the ellipse is described by the argument of perigee, ω, right ascension of the ascending node, ψ, and the inclination, ι. The position along the ellipse is described by the true anomaly, ν. The mean anomaly, m, is used instead of the true anomaly to represent the along-track position as it varies linearly over time. Aside from the mean anomaly, the orbital elements do not change over time for an ideal system, which makes it easy to compute the position of the satellite. The disadvantage of the Keplerian representation is that it becomes more difficult to include many of the complex perturbing accelerations.

In the present study, a special case of the orbital elements, the Brouwer Mean elements as realized in the SGP4 propagator, are employed to simplify the calculation of the largest of the orbital perturbations. The SGP4 propagator is an analytical model for propagating the orbit of a satellite forward in time. It is an amalgamation of two propagators, the original SGP4 and a deepspace version, the SDP4. As nanosatellites have thus far been restricted to low Earth orbits, the deepspace portion of the propagator is not needed (Hoots et al., 2004).

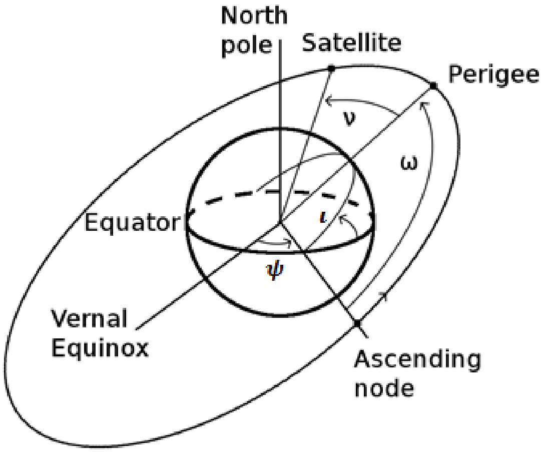

Figure 4-2 Definition of the Keplerian orbital elements.

The SGP4 propagator uses its own version of orbital elements to store the orbit of a satellite. It uses Two Line Elements, TLEs, which are based on Brouwer Mean orbital elements and a drag term. The Brouwer Mean orbital elements are an attempt to include perturbations due to the J2, J3, J4, and J5 (Hoots et al., 2004) terms for the spherical harmonic expansion of the Earth's gravitational field in the orbital model. These terms account for the oblateness of the Earth and the largest portion of the lack of symmetry between the Northern and Southern hemispheres. The inclusion of these terms in an analytical model allows higher accuracy orbit determination, while maintaining the ease and speed of orbit calculations that the orbital elements allow. The SGP4 propagator drops terms with eccentricity as the divisor, improving calculation speed with minimal loss in accuracy for satellites with small eccentricities. In addition to the first five spherical harmonic coefficients, the SGP4 includes an aerodynamic drag model that allows quick calculation of the decay of the satellite. It uses a power series model of the Earth's atmosphere with simplified equations at perigees lower than 220 km altitude. The drag is calculated using a density model and an empirically determined drag coefficient $b^{\star}$ (Hoots et al., 2004).

The SGP4 propagator deals with secular, short-term periodic, and long-term periodic perturbations. First, the secular perturbations are applied, then short-term and long-term periodic perturbations. In order to apply the SGP4 propagator to use with an EKF, the secular perturbations are separated from the short-term and long-term periodic perturbations. This separation allows the Brouwer mean elements to be kept current for a given epoch by applying the secular perturbations. The short-term and long-term periodic perturbations are only applied when the position of the satellite is required, as for generating the expected magnetic field from the IGRF.

NORAD tracks orbiting objects over 8 cm in diameter in low Earth orbit (Flury, 1995) and publishes TLEs over the Internet. TLEs include current Brouwer mean elements and the drag coefficient in an efficiently packaged form. Though, before the SGP4 algorithm can be applied, the provided mean motion must first be converted to a Brouwer mean semi-major axis (Hoots et al., 2004). Because the mean motion, as provided in the TLE, brings no advantages to the EKF, the filter uses the Brouwer mean semi-major axis exclusively. To make use of TLEs with the version of the SGP4 propagator created for this work, the Brouwer mean semi-major axis needs to be calculated separately (Vallado et al., 2006).

4.2 Implemented Orbit Determination EKF

To estimate the orbit, the standard EKF, described in Section 1.4, is used. It follows equations (1-1) through (1-10), and the same assumptions of the measurement and process noises, wherein the noises are assumed to be independent, zero mean, Gaussian processes that are uncorrelated in time. The state vector consists of the Brouwer mean orbital elements and the SGP4's drag coefficient $b^{\star}$.

$$\vec{x} = \begin{bmatrix} \alpha \\ \varepsilon \\ \iota \\ \omega \\ \psi \\ m \\ b^{\star} \end{bmatrix} \tag{4-4}$$

with units of radians for ι, ω, ψ, and m, and units of Earth radii for α. The Brouwer mean orbital elements were chosen as six parameters are needed to fully describe an ideal orbit, and they are the parameters used by the SGP4 propagator. The largest of the perturbing accelerations at low altitudes where nanosatellites orbit are the Earth's gravity terms to J5 or J6 and the drag. As the short-lived nanosatellites decay, the drag term becomes increasingly important (Montenbruck and Gill, 2005). The complete SGP4 parameterization, including drag coefficient, is used to account for the largest of the perturbing accelerations. To propagate the state forward through time, the secular perturbation portion of the SGP4 propagator is used as the dynamic model. The inclination and drag coefficient do not change when the dynamic model is applied.

4.2.1 Measurement Update of the Orbit Determination State Vector

The a priori measurement is calculated using the short-term and long-term period perturbations of the SGP4 propagator to calculate the position of the satellite in TEME coordinates. The position is then converted to ECEF coordinates by applying the sidereal time correction. To reduce complexity, the polar motion corrections are not applied. The ECEF position is used to generate the magnetic field in ECEF coordinates using the IGRF and equations (2-24) through (2-26).

Two methods are available for the measurement. The first is to use the complete three axes of the magnetic field measurement to update the state vector. The second is to use only the magnitude of the measured vector. The first method should provide more accurate orbit determination, however accurate understanding of the attitude knowledge is required. The advantage of the second method is that it is independent of the attitude knowledge, which makes it the logical choice when considering the coupling of this EKF with the attitude EKF, which generates inaccurate covariances during convergence. The measurement for orbit estimation becomes (Jung and Psiaki, 2002):

$$\vec{z}_O^- = h(\vec{x}^-) = \left|\vec{b}(\vec{x}^-)\right| = \sqrt{\vec{b}_x(\vec{x}^-)^2 + \vec{b}_y(\vec{x}^-)^2 + \vec{b}_z(\vec{x}^-)^2} \tag{4-5}$$

where $\vec{b}_x(\vec{x}^-)$, $\vec{b}_y(\vec{x}^-)$, and $\vec{b}_z(\vec{x}^-)$ are calculated from equations (2-24) through (2-26). This leads to the Jacobian:

$$C_t = \left.\frac{\partial h(\vec{x})}{\partial \vec{x}}\right|_{\vec{x}_t^-} = \left.\frac{\partial h(\vec{x})}{\partial \vec{b}}\frac{\partial \vec{b}}{\partial \vec{x}}\right|_{\vec{x}_t^-} \tag{4-6}$$

$$\frac{\partial h(\vec{x})}{\partial \vec{b}} = \begin{bmatrix} \frac{\vec{b}_x}{\vec{z}_O} & \frac{\vec{b}_y}{\vec{z}_O} & \frac{\vec{b}_z}{\vec{z}_O} \end{bmatrix} \tag{4-7}$$

The Jacobian $\frac{\partial \vec{b}}{\partial \vec{x}}$ is given in Appendix B. Given equation (4-5), the new measurement covariance matrix $R_{|\vec{b}|}$ can be calculated from the covariance of the sensor $R_{\vec{b}}$:

$$R_{|\vec{b}|} = \left[\frac{\partial h(\vec{x})}{\partial \vec{b}}\right] R_{\vec{b}} \left[\frac{\partial h(\vec{x})}{\partial \vec{b}}\right]^{\mathsf{T}} \tag{4-8}$$

$$R_{|\vec{b}|} = \begin{bmatrix} \frac{\vec{b}_x}{\vec{z}_O} & \frac{\vec{b}_y}{\vec{z}_O} & \frac{\vec{b}_z}{\vec{z}_O} \end{bmatrix} \begin{bmatrix} 200 & 0 & 0 \\ 0 & 200 & 0 \\ 0 & 0 & 200 \end{bmatrix} \begin{bmatrix} \frac{\vec{b}_x}{\vec{z}_O} \\ \frac{\vec{b}_y}{\vec{z}_O} \\ \frac{\vec{b}_z}{\vec{z}_O} \end{bmatrix} \tag{4-9}$$

$$R_{|\vec{b}|} = \left(\frac{\vec{b}_x^{\ 2}}{\vec{z}_O^{\ 2}} + \frac{\vec{b}_y^{\ 2}}{\vec{z}_O^{\ 2}} + \frac{\vec{b}_z^{\ 2}}{\vec{z}_O^{\ 2}}\right) 200 = 200 \tag{4-10}$$

The estimated state is updated using the standard EKF formulae from equation (1-5) and its covariance from equation (1-6).

4.3 Simulation Studies Used in Evaluating the Orbit Determination EKF

To demonstrate the performance of the orbit determination filter, several simulations were run. The simulations used the same typical nanosatellite orbit parameters used for the attitude determination simulations in Chapter 3 as it facilitates easier comparisons between the different EKFs. The initial conditions used a nearly circular, Sun synchronous, noon-midnight, orbit at an apogee height of 400 km, with a mean anomaly and argument of perigee of 0°, as shown in Table 4-1. More detail is presented on the variety and distribution of typical nanosatellite orbits in section 5.2. The orbit EKF is independent of attitude, so no attitude information was used in the tests.

4.3.1 Initialization of the State Vector for Orbit Determination Simulations

The EKF is designed to be used either immediately upon launch or after the upload of an initial TLE. Three separate cases were examined for initialization. The first case assumed a TLE had been uploaded and used to initialize the EKF. The second and third cases assumed the EKF would be used from launch, with the third case also assuming there had been some difficulty starting the satellite, and a large error was present in the mean anomaly. The initial conditions for all three cases are shown in Table 4-1.

To initialize the EKF, the design orbit was used with truncated values to represent the uncertainty in orbit insertion. Typical orbit insertion errors are on the order of those given in Table 4-2. The mean anomaly insertion error was calculated based on time to start the EKF from a powered off state. 1σ startup time was assumed to be 5 minutes.

Table 4-1 Initial state vectors used for orbit determination tests.

	True Values	Case a (low initial error)	Case b (high initial error)	Case c (extreme mean anomaly error)
$\boldsymbol{\alpha}$ **(Earth Radii)**	1.0627142	1.0627	1.06	1.06
$\boldsymbol{\varepsilon}$	0.0015	0.0015	0.005	0.005
$\boldsymbol{\iota}$ **(rad)**	1.6935733	1.69	1.69	1.69
$\boldsymbol{\omega}$ **(rad)**	0.0	0	0.02	0.02
$\boldsymbol{\psi}$ **(rad)**	1.7719979	1.77	1.77	1.77
$\boldsymbol{m}$ **(rad)**	0.0	0	0.1	1
$\boldsymbol{b^{\star}}$	–	0.001	0	0

Table 4-2 Approximate orbital insertion error on launch.

Orbital Parameter	**Insertion Error (3σ)**	**Insertion Error (1σ)**
Semi-major axis (Earth Radii)	0.009	0.003
Inclination (rad)	0.003	0.001
Eccentricity	0.009	0.003
Argument of Perigee (rad)	0.003	0.001
RAAN (rad)	0.003	0.001
Mean Anomaly (rad)	0.9	0.3

4.3.2 Simulation Results for the Orbit Determination EKF

Based on the results from Laneve (1997) and Abdelrahman (2011), the EKF should have enough information to converge to the correct orbit within ten orbits to an accuracy of approximately 40 km. Figure 4-3 (a) shows the initial convergence period of the three cases. Case ‘a’, starting with little initial error, attained steady state performance within 1 orbit and was the most accurate of the three cases. The filter converged quickly in both cases ‘b’ and ‘c’, taking only 4 orbits to reach a steady state in both cases. Even though case ‘c’ had much more error in the mean anomaly than case ‘b’, case ‘c’ took no extra time to converge as the observability of the under-determined system was the driving force in the convergence time. This quick convergence was well within what was expected, with the results of Laneve (1997) and Abdelrahman (2011) taking 10 and 15 orbits, respectively. Orbit EKF convergence in 4 orbits was faster than expected, but for extremely short missions could have some impact on the ability of a nanosatellite to perform its mission. For nanosatellites functioning for longer than a few months, the impact becomes less significant, as the 6 hour convergence period typically only needs to be performed once.

As all three cases used the same orbit parameters, they showed very similar structure in the steady state portion in Figure 4-3 (b). The similar time response in the three cases could be attributed to the unmodelled disturbances and identical orbits. Cases ‘b’ and ‘c’ had significantly larger peaks in the error as the initial conditions still affected the estimate.

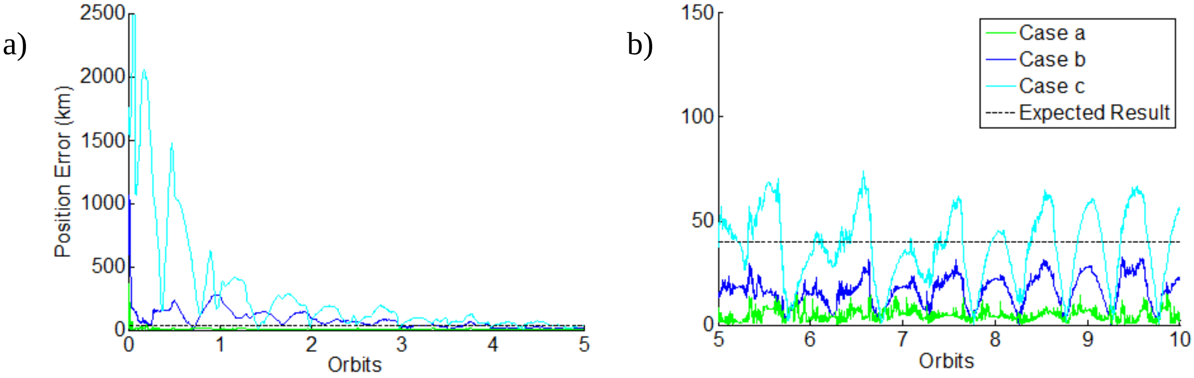

Figure 4-3 Orbit EKF response for all three cases, showing a) convergence, and b) steady state performance.

The steady state accuracy of case ‘a’ was expected to be the best, as less error was included in initialization. Case ‘b’ was expected to have larger steady state error than case ‘a’, as the initial estimates were much worse. Case ‘c’ was expected to have the largest steady state errors as the initial estimates were the worst. With a better initial estimate, a reduced steady state error was shown as is common in EKFs. Cases ‘a’ and ‘b’ showed steady state accuracies well above the 40 km that was expected, largely due to the accuracy of the initial state estimate. Case ‘c’, however, was worse than expected due to the inaccurate initial estimate. Wherever possible, care should be taken to provide accurate initial state estimates in order to keep the steady state error at a minimum. Table 4-3 shows the steady state accuracy of the orbit determination EKF in each case through the 95th percentile of the overall position and velocity. This accuracy serves as a baseline to which subsequent results can be compared.

Table 4-3 95th percentile of error in overall orbit estimates for all test cases.

	Case a	**Case b**	**Case c**
$\left\|\widehat{d\dot{\vec{r}}}\right\|$ **(km/s)**	1.11×10^{-2}	3.01×10^{-2}	6.94×10^{-2}
$\left\|\widehat{d\vec{r}}\right\|$ **(km)**	9.86	27.9	64.1

Although the filter converged satisfactorily, the estimated covariance failed to accurately reflect the accuracy of the estimated parameters during the convergence period, as seen in Figure 4-4. For this reason, the attitude and orbit filters were kept largely independent. Rather than updating the orbit with the full vector of the magnetic field, the orbit continued to be updated with the attitude independent magnitude of the magnetic field. As with the attitude determination filter presented in Chapter 3, the disagreement between the covariance and error during convergence was not unexpected because the filter is under-determined, and the covariances were updated based on the agreement between the measurements and the estimate. This discrepancy made it difficult to determine when the EKF had converged without the aid of a reference solution. It may be possible to reduce the discrepancy through tuning, or by using a smoothing estimator to identify convergence. Initializing the EKF with an estimate that is already accurate to better than 40 km, by utilizing TLEs, can also be used to eliminate the convergence time, as with case ‘a’.

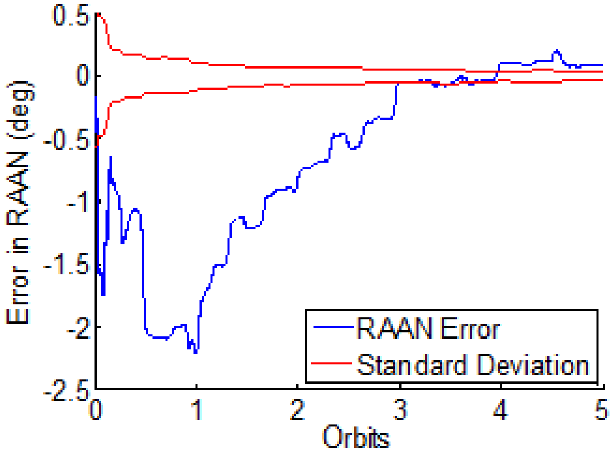

Figure 4-4 Representative discrepancy between covariance and estimated parameters during convergence from case b.

4.4 Summary of the Orbit Determination Extended Kalman Filter

An orbit determination algorithm based on an EKF was developed to estimate the Brouwer mean orbital elements and drag term of a nanosatellite. The orbital elements were propagated using the secular variations portion of the SGP4 propagator. The EKF accomplished the estimation by comparing the magnitude of magnetic field measurements to the IGRF. Three test cases have shown that the EKF created is able to handle orbit determination for nanosatellites autonomously with magnetometers. The EKF took only 40% of the expected 10 orbits to converge. The tests have set the baseline for orbit determination to be used for comparison with combined orbit and attitude determination at a convergence time of 4 orbits and an accuracy of 27.9 km. The EKF had difficulty identifying convergence, so it should only be used when it can be properly initialized, or as a backup to other systems, where TLEs or additional sensors can aid in initializing the EKF or determining convergence. Improvements may be possible with further tuning, or the utilization of a smoothing estimator to determine convergence.

5 Combined Attitude and Orbit Determination

To show the effects that magnetometer-based orbit determination has on attitude determination, and evaluate the effectiveness of an autonomous approach to attitude and orbit determination on nanosatellites using magnetometers, the combined EKF for orbit and attitude determination is presented. First, the interface between the EKFs from Chapters 3 and 4 is described. Second, the test cases are presented. Third, the results of the test cases are presented and compared to the baselines from Chapters 3 and 4. And lastly, the effectiveness of magnetometer-based autonomous attitude and orbit determination for nanosatellites is evaluated.

5.1 Interface Between the Attitude and Orbit Determination EKFs

To perform both the attitude determination and orbit determination simultaneously, two options are possible. One filter, including both attitude states and orbit states, could be created, or the two individual EKFs from Chapters 3 and 4 could be run concurrently. As the attitude determination and orbit determination are largely independent, creating one large EKF would use a significantly larger amount of computational power than running the two EKFs concurrently. To reduce the computational burden, the two Extended Kalman Filters (EKFs) are run concurrently, with dynamic and measurement updates happening one after the other. First, both dynamic updates are performed. Then, both measurement updates are performed. By performing both dynamic updates first, a significant amount of the calculations are performed ahead of time, so the measurement update can happen quickly, as soon as a measurement is taken. As the two dynamic update algorithms are independent, neither using information from the other, it does not matter which is updated first. For the tests performed, the attitude is updated first, followed by the orbit.

For the measurement update, the two filters are not independent. The measurement update in the orbit determination filter is independent of attitude as only the magnitude of the vector is used. However, the measurement update of the attitude filter depends significantly on the orbit as the position of the satellite is necessary for generating the correct a priori magnetic field vector, both in magnitude and direction. For this reason, the orbit is updated first, followed by the attitude.

The two filters used are identical to those used in the independent cases, except for one detail. To ensure that an inaccurate position does not lead to an over-emphasis on the alignment of an incorrect magnetic field vector, the measurement needs to be de-weighted appropriately. The effect of the uncertainty in the orbit on the uncertainty in the magnetic field vector is, from the error propagation equation (Ochoa and Belongie, 2006):

$$P_{\vec{b},O} = \left.\frac{\partial \vec{b}}{\partial \vec{x}_O}\right|_{\hat{\vec{x}}_O} P_O \left.\frac{\partial \vec{b}}{\partial \vec{x}_O}\right|_{\hat{\vec{x}}_O}^{\top} \tag{5-1}$$

which leads to the uncertainty in the misclosure being:

$$P_d = R + \left.\frac{\partial \vec{b}}{\partial \vec{x}_O}\right|_{\hat{\vec{x}}_O} P_O \left.\frac{\partial \vec{b}}{\partial \vec{x}_O}\right|_{\hat{\vec{x}}_O}^{\top} + C_A P_A C_A^{\top} \tag{5-2}$$

and the Kalman gain being:

$$K = P_A C_A^{\top} P_d^{-1} = P_A C_A^{\top} \left(R + \left.\frac{\partial \vec{b}}{\partial \vec{x}_O}\right|_{\hat{\vec{x}}_O} P_O \left.\frac{\partial \vec{b}}{\partial \vec{x}_O}\right|_{\hat{\vec{x}}_O}^{\top} + C_A P_A C_A^{\top} \right)^{-1} \tag{5-3}$$

The independence of the added uncertainty term from the estimated attitude parameters allows the added uncertainty to be thought of as uncertainty in the measurements.

5.2 Simulation Test Cases Used in Evaluating the Combined EKF

To demonstrate the performance of the combined EKF, five test cases are presented. The purpose of these tests was to show the EKF's performance over the range of nanosatellite orbits. Nanosatellites typically occupy orbits with an apogee altitude between 950 km and 150 km, with the majority of these above 550 km as shown in Figure 5-1 (a). (b) shows the distribution of orbital inclinations, with the vast majority at Sun synchronous inclinations of approximately 100°. Inclinations down to 30° are also used, though less frequently. The eccentricities of nanosatellites, shown in (c), provide mostly circular orbits. The eccentricities vary between 0.0005 and 0.08, with the majority at approximately 0.0015. RAAN is not included as its physical meaning changes over time. Argument of perigee is not included as it has less impact as most of the orbits are nearly circular. Figure 5-1 also includes the distribution of selected test cases as comparison.

The test cases were designed to cover the extent of the various nanosatellite orbits as shown in Figure 5-1 as well as the most common characteristics of the orbital elements. The results presented in section 5.3 are dependent on the chosen initial conditions. While the initial conditions are chosen to cover the majority of nanosatellite missions, different results may be observed given different test conditions, such as lower inclination angles. The initial conditions for the five test cases are presented in Table 5-1. Unlike the previous nanosatellite orbits, the test cases did not include any orbits with apogee altitudes lower than 400 km as they decay within several months and the long convergence period of the orbit determination EKF could detract from the time that could be devoted to operating the payload.

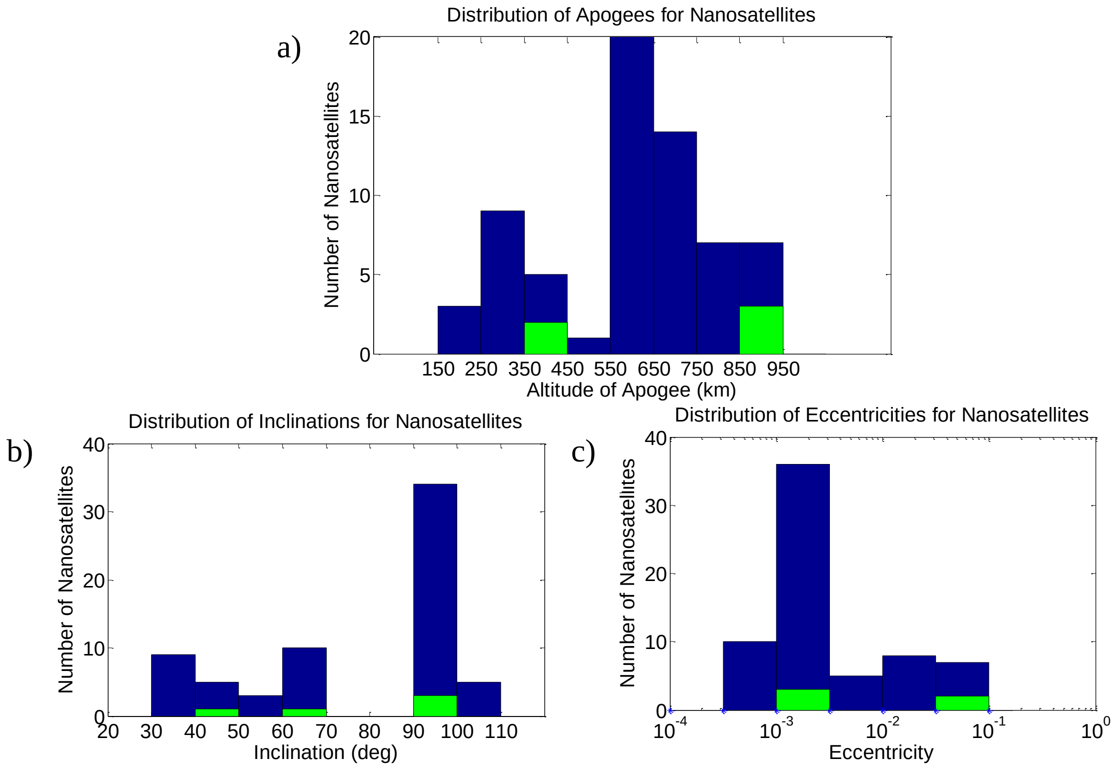

Figure 5-1 Distribution of orbital elements for previous nanosatellite missions, with previous missions in blue and the test cases in green. a) altitude of apogee b) inclination c) eccentricity

Most of the test cases were chosen to be Sun synchronous, with two cases spanning the lower inclinations from Figure 5-1. The Sun synchronous orbits were split between dawn-dusk, favoured for the added power generation due to avoiding the zones of eclipse, and noon-midnight, with the RAAN set appropriately. Most of the eccentricities coincided with the majority of actual nanosatellite orbits. The remaining two cases encompassed the extents of eccentricities seen in nanosatellites, as well as demonstrating the effects of non-circular orbits. The altitudes were chosen to cover the extents of the various orbits seen in nanosatellites with apogee altitudes over 400 km. The eccentricity, RAAN, and argument of perigee of cases 3 and 4 were chosen to coincide with those of actual nanosatellites with lower inclinations. Test case 1 was chosen to coincide with the baseline cases used in Chapters 3 and 4, and was examined under two initial estimates. Case ‘1a’ used the same initial estimates as case ‘b’ from both Chapters 3 and 4, while case ‘1b’ used the same initial estimates as case ‘b’ from Chapter 3 and the large initial mean anomaly error from case ‘c’ in Chapter 4. All orbits were simulated as starting with a mean anomaly of 0. The initial conditions for cases 2 to 5 were generated the same way as those in case ‘1a’.

Table 5-1 Summary of initial conditions.

Test Case	Eccentricity	Inclination (°)	Apogee Height (km)	RAAN (°)	Argument of Perigee (°)	Notes
1	0.0015	97.0346	400	101.528	0	Sun synchronous, noon-midnight
2	0.0015	99.0383	900	191.528	0	Sun synchronous, dawn-dusk
3	0.08	60	900	237.7268	43.7241	
4	0.00118	40	400	0.06	217.015	
5	0.08	99.0383	900	191.528	0	Sun synchronous, dawn-dusk

5.3 Results of Combining Attitude and Orbit EKFs

The combined EKF showed varying convergence periods for the different tests. The attitude portion always converged faster than the orbit portion. The attitude portion generally took 1 orbit, while the orbit portion generally took approximately 8 orbits. While the 1 and 8 orbit convergence times were longer than those shown in Chapters 3 and 4, respectively, it was still well within that reported by Abdelrahman (2011). Convergence periods for the various tests are shown in Table 5-2. The attitude portion took slightly longer to converge than the 0.5 orbits taken when estimated alone. The longer convergence time was due to the coupling of the two filters, and was evident in the greatly increased convergence time in case ‘1b’. While the orbit filter still had large amounts of error, the attitude could not compare the measured magnetic field to an appropriate estimate from the IGRF. As the magnetic field only varies slowly with on orbit position, errors of a few hundred kilometres make little difference to the attitude filter. If the orbit filter was given an accurate initial estimate, the attitude filter would converge more quickly. Unlike the attitude portion, the convergence of the orbit portion was independent of the attitude, as only the magnitude of the magnetic field was used, and was consistent with the 4 orbits taken when estimated alone. The variability in convergence times did not correlate well with the design orbits, and was likely due to the choice of initial conditions, though the altitude of apogee may have had some effect, as the lower orbits appeared to converge more quickly.

Table 5-2 Convergence periods for the 5 test cases.

Test Case	**Attitude Convergence Time (orbits)**	**Orbit Convergence Time (orbits)**
1a	0.6	5
1b	2	11
2	1	8
3	1	8
4	0.8	7
5	0.9	9

The convergence period of the inclination for test case 1b, shown in Figure 5-2, was typical of the other cases, with convergence times as listed in Table 5-2. The initial jump seen in this case was caused by the EKF compensating for the large amount of initial error in the mean anomaly. Some of this jump may be able to be compensated for with constraints based on the initial conditions, or by tuning the initial state covariance matrix. As with the attitude and orbit EKFs developed in Chapters 3 and 4, the covariances of the state vector did not reflect the level of convergence of the filter. It may be possible to reduce the discrepancy through tuning, or by using a smoothing estimator to identify convergence. Initializing the EKF with an estimate that is already accurate to better than 40 km and 5°, by utilizing TLEs and Sun sensors, can also be used to eliminate the convergence time.

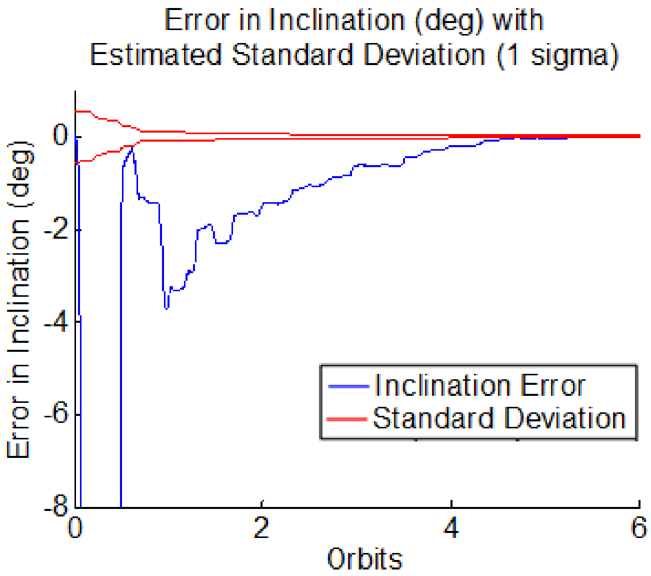

Figure 5-2 Convergence period for inclination with covariance in test case 1b. Covariance is in red, error in inclination is in blue.

For the steady state, however, the covariance did reflect the accuracy of the estimated terms. The covariance of the estimated terms was not in full agreement with all of the terms, but was within an order of magnitude. More tuning of the EKF is required to bring complete agreement between the covariance estimated by the EKF and the standard deviation of the terms.

The 95th percentile of the position and velocity for the 5 test cases is given in Table 5-3. The time before steady state statistics were calculated was longer for these test cases than with the orbit-only EKF, because the estimate continues to improve over time and the combined tests were run for much longer. The delay in measuring steady state statistics led to misleading improvements in the steady state statistics with comparison to the baseline developed in Chapter 4. The steady state statistics for the orbit portion of the combined EKF were comparable to the offset initialization of the orbit-only EKF, and in some cases closer to the accurate initialization. The improvement was due to the longer run times of the combined filter, which continued to refine the estimates after convergence. Even case ‘1b’, with a large initial error in the mean anomaly, eventually converged to an accuracy consistent with case ‘1a’ and the accurate initialization from Chapter 4.

Table 5-3 95th percentile in estimated orbit error.

Test Case	**1a**	**1b**	**2**	**3**	**4**	**5**
$\left\|d\hat{\dot{\vec{r}}}\right\|$ **(km/s)**	1.25×10^{-2}	1.23×10^{-2}	1.35×10^{-2}	1.43×10^{-2}	1.49×10^{-2}	1.24×10^{-2}
$\left\|d\hat{\vec{r}}\right\|$ **(km)**	10.96	10.87	13.44	14.07	13.147	12.31

The choice of orbital elements as the estimated parameters came with one drawback, nearly circular orbits led to a coupling between the mean anomaly and argument of perigee. The coupling was evident in all cases except 3 and 5, where the increased eccentricity led to more accurate estimations of the argument of perigee and mean anomaly. The reduced error in these two parameters did not, however, lead to an increase in the accuracy of the resultant position and velocity in the TEME frame. As shown in Figure 5-3, the estimated mean anomaly and argument of perigee had equal and opposite biases in case 2, which had a small eccentricity. With a circular orbit the coupling between the two values led the equal and opposite biases to cancel out and put the satellite in very close to the same position.

Based on the results of Psiaki (1990) and Abdelrahman (2011), performing simultaneous attitude and orbit determination was expected to cause a loss in attitude knowledge accuracy of several degrees due to the loss in accuracy of the orbit knowledge. The 95th percentile of the total attitude error is given in Table 5-4. The combining of the two EKFs caused a twofold decrease in the accuracy in every test except case 4. The loss of accuracy of attitude estimates was due to the reduced accuracy of the position estimates in the combined EKF. The reduced position accuracy in turn created a reduced accuracy in the a priori measurement estimate. Improvements in the steady state attitude error can be obtained by including other sensors, or including disturbance torques in the dynamic model. Case 4, being the case with the smallest inclination and altitude, had the greatest attitude knowledge, similar to that found in the attitude-only EKF, which used perfect position knowledge. The other case with low altitude did not show significant improvement over the other test cases, implying that the cause of the improvement may have been the low inclination angle. The magnetic field vector likely did not change as much with errors in position at lower inclinations, leading to this increase in attitude knowledge. However, inclinations close to 0° are likely to lead to difficulty obtaining the third axis of attitude knowledge as the magnetic field direction does not vary as much.

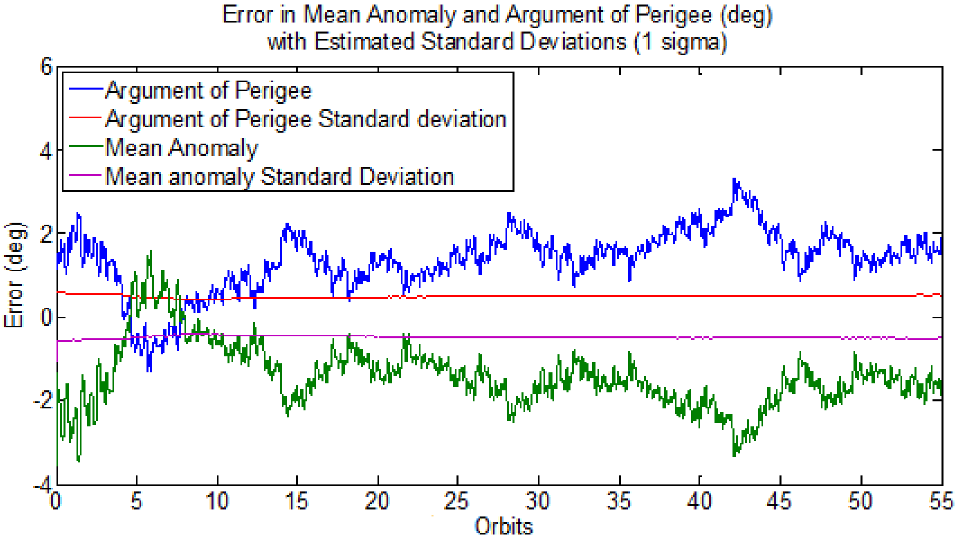

Figure 5-3 Comparison between Mean Anomaly and Argument of Perigee from test case 2.

Both the orbit portion and the attitude portion had unmodelled error sources included in the measurements. These showed up in the estimated parameters. In particular, much of the unmodelled error was collected in $b^{\star}$ for the orbit portion of the EKF and the three angular rate terms in the attitude portion of the EKF, as evidenced by the biases and erratic nature of the error in the estimates of these parameters. The erratic nature and biases, shown for $\vec{\omega}_z$ from test case ‘1b’ in Figure 5-4, were typical of the remaining angular rates as well as $b^{\star}$. In order to reduce the biases and make the covariances more closely match the accuracy of the estimated parameters, additional disturbances need to be included in the dynamic models, and the EKFs need to be tuned to account for the new models.

Table 5-4 95th percentile of error in overall steady state attitude estimate.

Test Case	**1a**	**1b**	**2**	**3**	**4**	**5**
Attitude knowledge error (°)	9.34	7.91	8.48	10.60	5.31	9.22

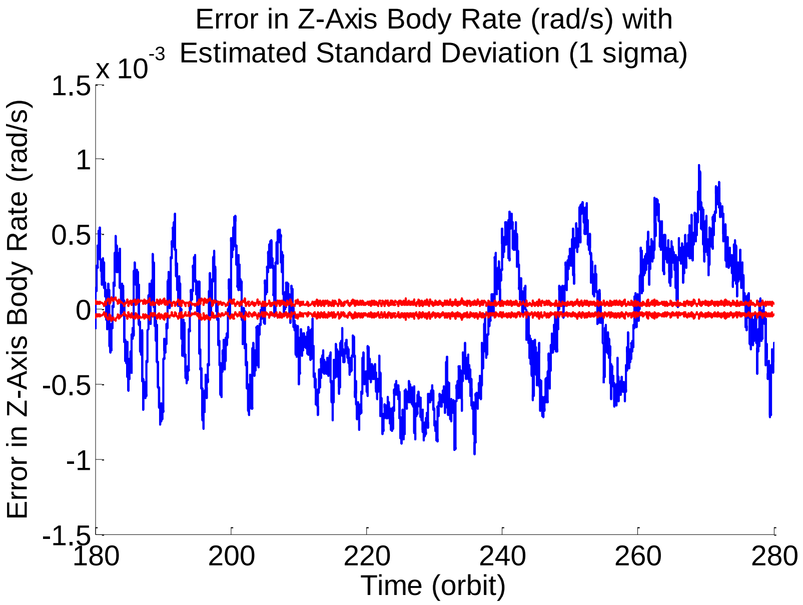

Figure 5-4 Typical bias and erratic behaviour in angular rate from test case 1b. Angular rate is blue, covariance is red.

5.4 Evaluation of Simultaneous Orbit and Attitude Estimation

Test case ‘1b’ showed convergence under adverse conditions, namely those of a highly inaccurate initial mean anomaly, with error of 1 radian (57.3°). It was run for 27 days to show the long-term stability of the developed EKF. The added time did not show significant improvement of the steady state error after many days, but it did accomplish the desired task of demonstrating long-term stability. Figure 5-5 shows the long-term plots of the position error and the overall attitude error.

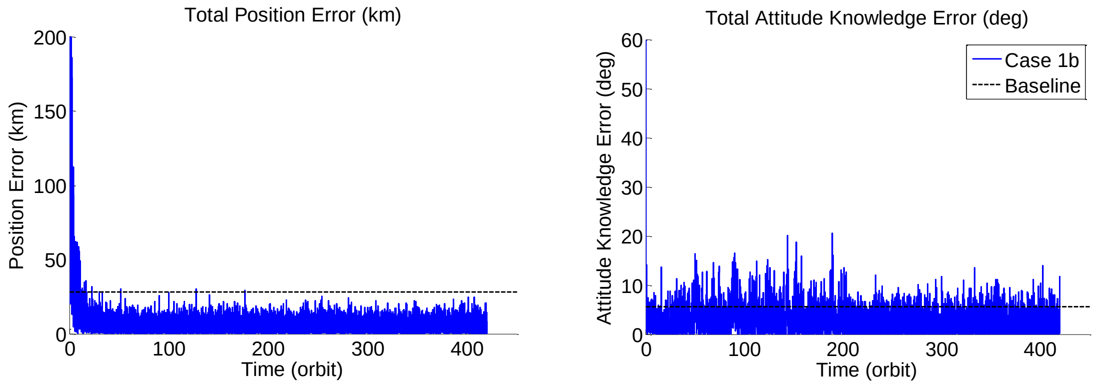

Figure 5-5 Long term stability of orbit and attitude determination EKF from test case 1b.

The orbital accuracy achieved was comparable to the accuracy of a TLE from NORAD after approximately 97 orbits, or 6 days. The orbital accuracy of TLEs used as comparison was based on the 95th percentile of NORAD TLEs for AAU CUBESAT (NORAD ID 27846). The satellite had an apogee altitude of 821 km, in a circular (eccentricity of 0.012), Sun-synchronous (inclination of 98.7°) orbit in the TLEs used. Three outlying TLEs, which did not agree with the TLEs before and after to under 5 km in the ECI frame, were not included in the dataset. Of the remaining TLEs, 95% had better agreement with the future TLEs than the one in Figure 5-6. After 6 days (90 orbits), the 95th percentile of the EKF orbit measurements outperformed the TLE in Figure 5-6.

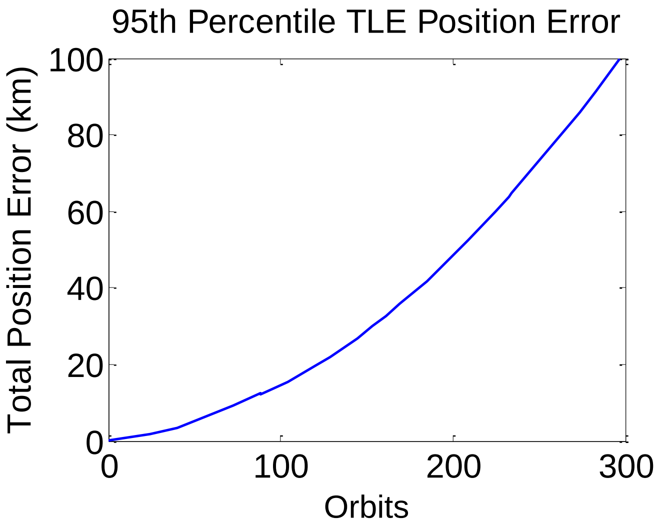

Figure 5-6 95th percentile NORAD TLE position error.

5.5 Summary of the Combined Attitude and Orbit Determination EKF

The two EKFs from Chapters 3 and 4 were combined by running both dynamic updates, then both measurement updates concurrently. To account for the effect the orbit has on the measurement uncertainty, an additional term, calculated from the orbit knowledge, was added to the measurement noise in the attitude determination portion of the combined EKF. Five test cases were used to show the performance across the range of typical nanosatellite orbit conditions. The test cases showed the effects magnetometer-based orbit determination has on attitude determination to be a loss in accuracy of nearly 4°. Combining the EKFs had no effect on the orbit determination. Convergence stayed largely the same, with a slight delay in attitude convergence due to a lack of orbit knowledge.

In evaluating the applicability of the algorithm to nanosatellite missions in general, several mission requirements were determined. For attitude determination, the degradation in attitude knowledge and difficulty identifying convergence require other sensors or an improved dynamic model to be used for any applications needing greater than 10° accuracy. This accuracy may make it difficult to use the algorithm for active control. The accuracy of the orbit determination makes this algorithm useful in any situation where TLEs are likely to be updated less than once a week. Accuracy of better than 15 km is good enough for most nanosatellite operations, including scheduling and pointing of instruments with fields of view larger than a few degrees. Fine knowledge of measurement locations would be better done with GPS or frequent updates of TLEs, if the mission allows. Due to the discrepancy between the covariances and state vectors, the algorithm is best used as a backup to other systems, so that convergence can be determined ahead of time.

6 Hardware-In-The-Loop Simulation

In order to validate the combined attitude and orbit determination EKF presented in Chapter 5 under realistic hardware noise conditions, a Hardware-In-The-Loop (HITL) test setup, designed to create and measure difficult to quantify errors from the satellite bus, is used. A Helmholtz cage is also presented, and its ability to simulate on orbit magnetic field conditions for ground testing is evaluated. First, the validation procedure is described. Second, the test setup is described. Third, the Helmholtz cage is described. Finally, the results of the HITL and Helmholtz cage tests are presented and evaluated in comparison to the results presented in Chapter 5.

6.1 HITL Test Procedure

To test the algorithm with the inclusion of typical error sources on nanosatellites, it is necessary to get measurements of the noise caused by the satellite bus and magnetometer. To this end, the satellite analog on the air bearing table, described in detail in section 6.2.2, is employed. Measurements are taken with the magnetometer through the satellite analog on the air bearing table. This measurement noise is then added to the expected magnetic field values, giving a realistic noise profile to use in testing the EKF. To evaluate the applicability of the Helmholtz cage to HITL attitude and orbit determination ground tests, additional tests are run with on orbit magnetic field values provided by the Helmholtz cage. Tests involving the Helmholtz cage are described in section 6.3.

To gather the necessary data, the air bearing table is placed at the centre of the cage. It is powered through a USB connection to a computer, and an Ethernet connection allows communication between the OBC and computer. The air bearing table is fixed in place to allow the background magnetic field to be removed and to allow comparison of the measured

Helmholtz cage output to the actual measurements. The xBee radio is not needed, as the tests are run with the air bearing table fixed in place. With the air bearing table set up, measurements of the magnetic fields associated with the satellite bus are taken. The background magnetic field is removed from the measurements, and the remaining noise is added to the expected measurements. These measurements are then processed by the EKF. The errors in the estimated position from the HITL tests are compared to those using simulated data to show their agreement, and lend validity to the simulated results.

6.2 HITL Test Apparatus

The HITL test setup is composed of an air bearing table with satellite analog, and a custom-built, three-axis Helmholtz cage. The air bearing table, which could be allowed to spin freely, is used to provide the satellite analog. It is placed in the uniform region of the Helmholtz cage. The Helmholtz cage is used to augment the measurements by providing a three-axis magnetic field similar to that observed on orbit. The Helmholz cage is only used for tests designed to evaluate its usefulness for ground testing.

To remove errors caused by stray magnetic fields from the satellite bus, larger satellites often place magnetometers on long booms (Acuña, 2002). However, the majority of nanosatellites are unable to employ these techniques, due to mass and volume constraints, so it is important to include them in the simulation. While the magnetometer has undergone testing to determine the distribution of the measurements and their standard deviation, any errors that may have been overlooked are included by using a typical nanosatellite magnetometer on the satellite bus.

6.2.1 Air Bearing Table Used in HITL Tests

The air bearing system, developed by Ustrzycki (2011), is a three-axis frictionless platform which allows ground testing of attitude algorithms. It uses a cup and ball arrangement with air nozzles to provide near frictionless rotation. It allows 360° rotation about the z-axis and 30° rotation about the x and y axes.

The air bearing platform is primarily used for attitude control tests, however, it provides a platform which already includes the satellite analog necessary for inclusion of the satellite bus sources of stray magnetic fields. The air bearing table has an air supply that lasts less than 1 hour. It also lacks an accurate system for determining its absolute attitude. For these reasons, only the satellite analog is used in the tests.

6.2.2 Satellite Analog Used in HITL Tests

The satellite analog consists of a power regulation system, an embedded computing system, an ADCS, and a communications system as shown in Figure 6-1. Power is supplied through USB from a computer. The power regulation system provides power to run the other subsystems at 3.3 and 5 V after conversion from the supply voltage (Ustrzycki, 2011). The communications system uses an xBee radio (Ustrzycki, 2011). It is powered, but not used for the tests. The embedded computing system uses a linuxstamp2 to do all the necessary computations, similar to what has been developed for nanosatellites by Borschiov (2012) in a different form factor. The ADCS board is a prototype board designed to be flown on a CubeSat™ in the QB50 mission. It contains all the sensors that would be used in flight, as well as the magnetorquer drive electronics (Li et al., 2013). The magnetorquers are orthogonally mounted inside a 1U CubeSat™ structure on the air bearing platform. All the subsystems are mounted outside the CubeSat™ structure for ease of access.

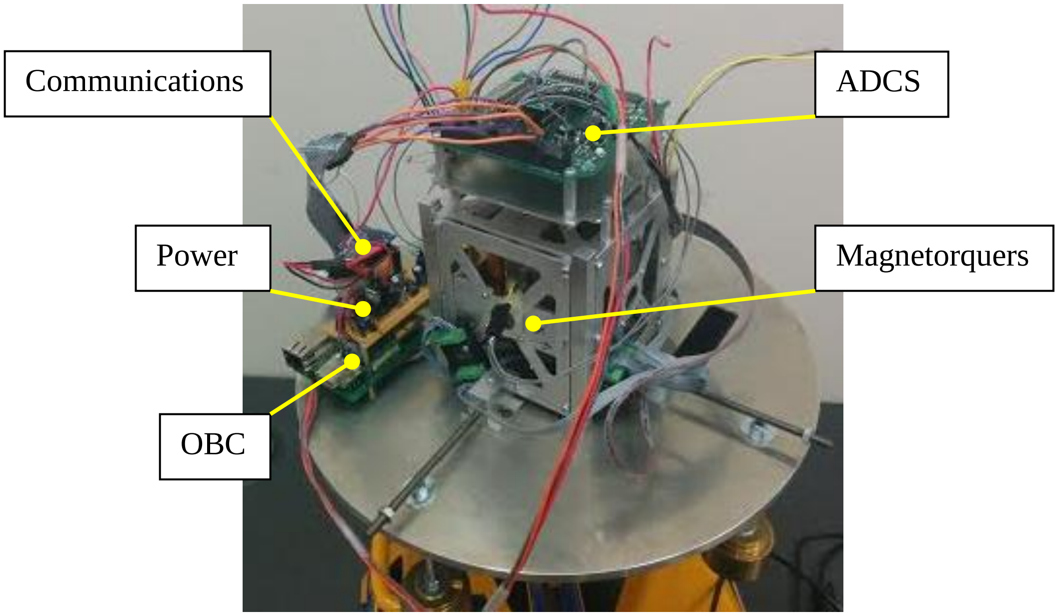

Figure 6-1 Satellite analog used for HITL tests.

6.2.2.1 Magnetorquers Used on the Satellite Analog

One of the major sources of bias in the magnetic field measurements comes from magnetorquers. Magnetorquers can be a source of both hard and soft iron magnetic fields. Hard iron magnetic fields being those caused by a magnetic dipole. Soft iron magnetic fields being those caused by ferromagnetic response of a material to outside magnetic fields. In their active state, magnetic torque rods act as hard iron sources. In their degaussed state, they act as soft iron sources.

The torque rods designed for the air bearing table's satellite analog are sized appropriately for a nanosatellite on orbit. The sizing is accomplished using equations (6-1) and (6-2) for total magnetic moment of a cored solenoid (Mehrjardi and Mirshams, 2010):

$$d = \frac{r_C e}{2\varrho_W}\left[1 + \frac{\mu_R - 1}{1 + (\mu_R - 1)n_D}\right] \tag{6-1}$$

$$n_D = \frac{4\left[\ln\left(\frac{l_C}{r_C}\right) - 1\right]}{\left(\frac{l_C}{r_C}\right)^2 - 4\ln\left(\frac{l_C}{r_C}\right)} \tag{6-2}$$

where the wire's resistance per unit length, ϱ_W, is related to resistivity:

$$\varrho_W = \frac{\rho}{a} = \frac{\rho}{\pi r_W^2} \tag{6-3}$$

These equations are evaluated for several different core sizes and generate Figure 6-2. The figure shows increasing magnetic moment with decreasing core radius. The power consumption for a given coil shows a decrease with increasing wire length and decreasing wire radius. The magnetorquer shows a reduction in weight with a decrease in core diameter and wire radius. The minimum available core and wire radiuses are chosen to maximize magnetic moment, and minimize power consumption and weight, respectively. The manufactured torque rods have a core radius of 2.86 mm and wire size of 36 AWG (Li et al., 2013).

For the HITL test the magnetorquers are placed in a degaussed state to minimize the effect they have on the magnetic field measurements. The reduction in magnetic field from degaussing is shown in Figure 6-3, in one axis. The degaussed state is used on orbit to minimize the effect on the measurements. Degaussing is accomplished by alternating the direction of current and reducing the current through the coils after each change of direction. The coils are energized long enough for the magnetic field to switch direction, but not long enough for the new magnetization to fully set in.

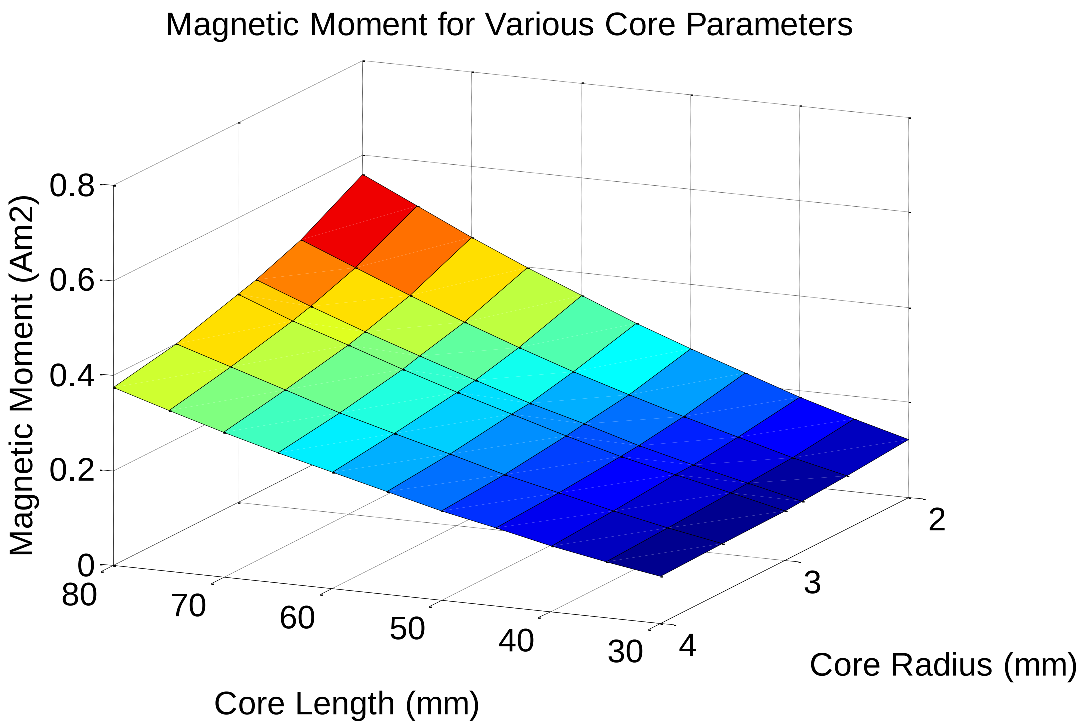

Figure 6-2 Magnetorquer sizing surface.

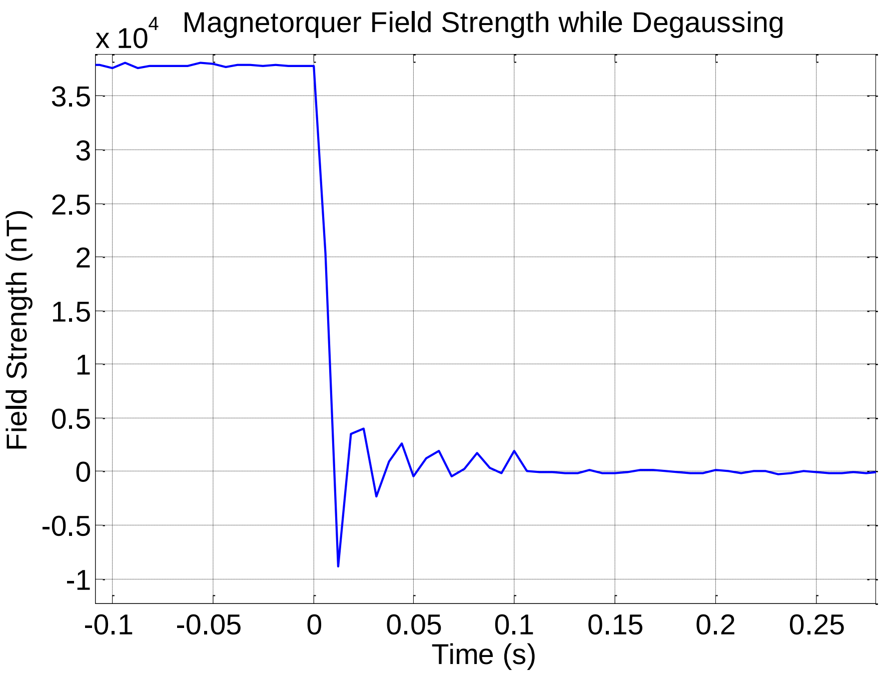

Figure 6-3 Magnetic field due to magnetorquers while degaussing.

6.2.2.2 Magnetometer Used on the Satellite Analog

The magnetometer used on the satellite analog is the same one assumed for the simulations in earlier Chapters. The magnetometer is a three-axis AMR magnetometer, and needs to be calibrated to remove orthogonality errors. The calibration ensured the obtained data could be properly decomposed into the three orthogonal axes. A series of measurements was gathered of a static magnetic field in as many orientations as possible to provide a full ellipsoid, shown in Figure 6-4. Biases and errors in orthogonality prevented the measurements from being of equal magnitude and centred about the origin. The transformation between this ellipsoid, in blue, and that of a sphere centered on the origin, in red, removed the bias and orthogonality errors and was obtained through the use of the Merayo technique by way of open source software (Barraud, 2009).

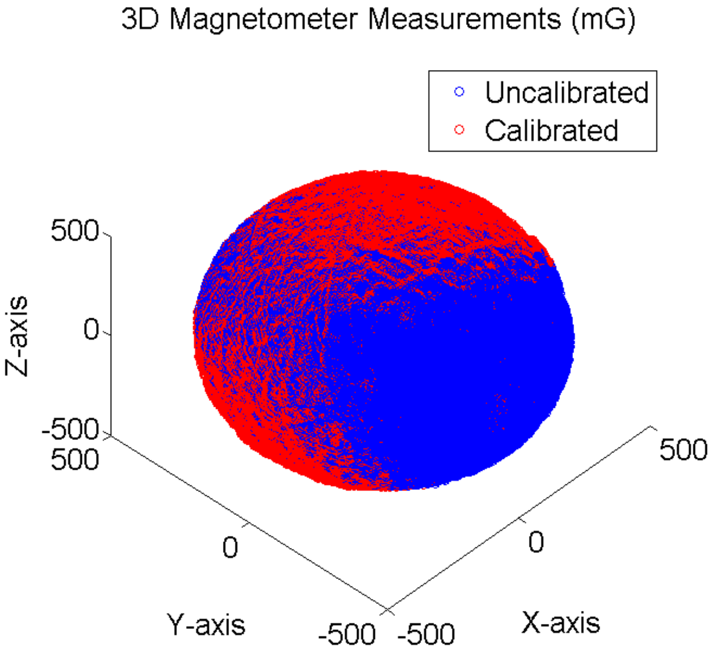

Figure 6-4 Comparison between calibrated and raw magnetometer measurements. Raw measurements in blue, calibrated measurements in red.

This transformation obtained through the Merayo technique was then applied to the measurements, using equation (6-4), to ensure orthogonal measurements were used in the EKF. The resultant transformation matrix, U, is given in equation (6-5), and the bias vector, c, is given in equation (6-6).

$$\vec{z}_{CAL} = U(\vec{z}_{RAW} - \vec{c}) \tag{6-4}$$

$$U = \begin{bmatrix} 1.0225 & 0.0161 & 0.0265 \\ 0 & 0.9723 & -0.0056 \\ 0 & 0 & 1.0261 \end{bmatrix} \tag{6-5}$$

$$\vec{c} = \begin{bmatrix} -243 \\ -1620 \\ 711 \end{bmatrix} \tag{6-6}$$

with $\vec{z}_{CAL}$ as the calibrated measurement in nT, and $\vec{z}_{RAW}$ as the raw measurement in nT. The calibrated measurement can now be used in the EKF without adding additional biases and attitude errors due to misalignment of the sensing elements.

6.2.2.3 Satellite Bus Used on the Satellite Analog

Another source of stray magnetic fields is the satellite bus. Current flowing in the printed circuit boards generates magnetic fields. These can add noise and bias to the measurements. For this reason, the ADCS board is included on the satellite analog, as well as close approximations of other subsystems.

The two subsystems that are not included are the science payload, which varies strongly from mission to mission, and the solar panels. The solar panels are not included as they would not be lit accurately, and so would not generate realistic magnetic field anomalies. Also, they would greatly complicate the assembly of the nanosatellite analog.

6.3 Helmholtz Cage Testing

A Helmholtz cage is used to create on orbit magnetic field conditions. Tests are run to determine if the magnetic fields generated are accurate enough for simultaneous attitude and orbit determination on the air bearing table. The air bearing table is aligned with the Helmholtz cage axes during Helmholtz cage tests to facilitate comparison with the expected magnetic field values. The Helmholtz cage is programmed to run through a set of measurements corresponding with the magnetic field measurements on orbit in the TEME frame. Measurements of the generated magnetic field are taken while the Helmholtz cage is on. The measurements are then aligned with the start of simulation, and the associated time adjusted accordingly. The three-axis measurements are then processed by the EKF.

The three-axis Helmholtz cage used to create on orbit magnetic field conditions is shown in Figure 6-5. By selecting the correct geometry and location of coils, a uniform magnetic field is created along the axis of one pair of coils. Three orthogonal sets of coils are used to cancel out the existing magnetic field in the central volume of the cage and apply a field in any desired direction in its place (Post et al., 2013).

At the time of experimentation, the y-axis coils were in need of repair, so only two of the axes could be evaluated. The y-axis measurements were used to add realistic noise to simulated y-axis measurements in the tests, as is done with the tests without the Helmholtz cage.

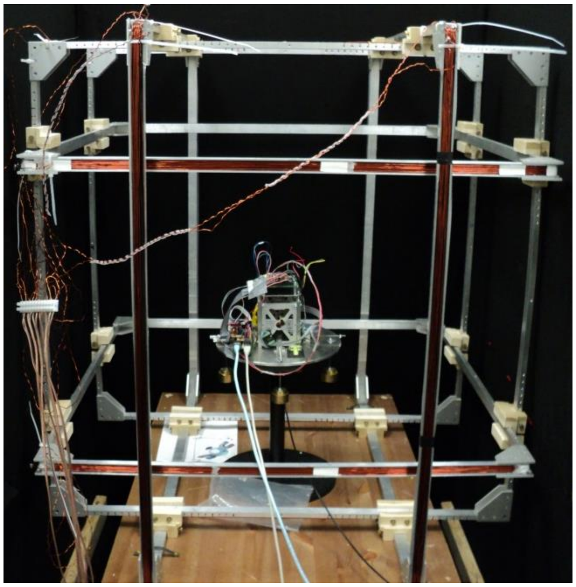

Figure 6-5 Helmholtz cage and air bearing table aligned and ready for testing.

6.3.1 Calibration of the Helmholtz Cage

To ensure the Helmholtz cage was functioning properly and giving reliable output, it needed to be calibrated. This involved placing a magnetometer in the uniform region of the Helmholtz cage, aligning it to the cage axes, and running the drive electronics through a complete cycle of possible outputs. The procedure was completed for all three axes in turn.

The drive electronics consist of a microcontroller driving a pulse width modulated, PWM, signal to h-bridges, which in turn drive the coils, as shown in Figure 6-6. The current from the h-bridge passes through a simple low-pass filter before entering the coil, and subsequently passing through another low-pass filter and returning to the other half of the h-bridge. The symmetrical design allows current to be passed in either direction, with the same functionality, allowing both positive and negative field values to be generated in each axis. Each coil is individually controllable by PWM with its own h-bridge and low-pass filters. Each axis is also powered from a separate power supply to limit cross talk between the coils. The drive electronics allows 128 different drive settings (including 0 current) and two direction settings (Post et al., 2013).

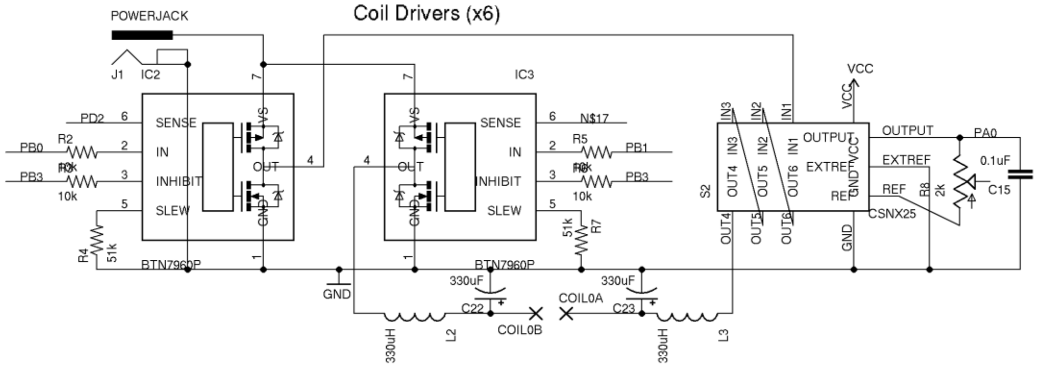

Figure 6-6 Helmholtz cage coil driver schematic (Post et al., 2013).

As the generated magnetic field of the coils shows significant noise (between 200 and 400 nT) when driven by the drive electronics, several field measurements were taken at each output setting and averaged. The averaged measurements were then plotted against the control setting. Examining the plot revealed a roughly bi-linear structure in the output, as is shown in Figure 6-7. Very large and very small values for the control setting were practically indistinguishable from the settings around them, requiring the settings to be changed to ensure that a binary search algorithm could properly process the data. Any data points which were not in ascending order were set to those of their nearest neighbour. The offset when the coils were not driven was also removed from the dataset, which allows a single binary search algorithm to process both the positive and negative output of the coils. The resultant calibration curve is shown for the x-axis in Figure 6-7. This procedure was repeated for the other functional axis, the z-axis, yielding similar results.

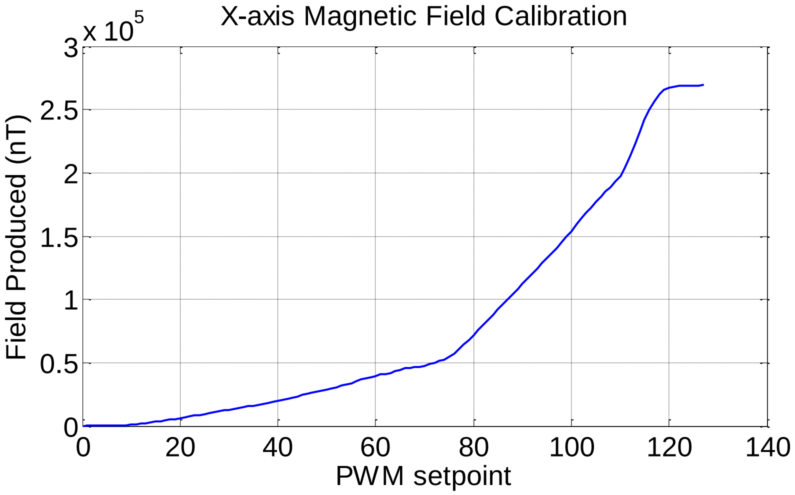

Figure 6-7 Magnetic field calibration curve showing magnetic field generated by the coils for various PWM duty cycles.

Drive control was maintained by taking the input of a desired magnetic field measurement, normalizing it to the background magnetic field, taking the absolute value, and applying the binary search algorithm to find the appropriate PWM setting. The sign was then re-applied to the PWM setting. The signed setting was then sent to the microcontroller driving the coils. The calibrated Helmholtz cage allowed two axes of on orbit measurements to be simulated so real measurements could be made with the desired hardware. The calibration enabled the suitability of the Helmholtz cage for simulating measurements to be examined. The measured accuracy of the Helmholtz cage is shown in Table 6-1. As the magnetometers used on the satellite analog are accurate to 200 nT, a loss in accuracy may be caused by the large magnetic field noise and control resolution. In particular, a control resolution of 4,000 nT may be too much for the attitude and orbit determination EKFs to compensate for. In order to compensate for this in software, the magnetic field measurements would need to be limited to the fields the cage is known to produce, which would invalidate the purpose of using the cage to produce on orbit magnetic field conditions for the testing of attitude control software.

Table 6-1 Helmholtz cage operating parameters (Post et al., 2013).

	Value	Units
Field Range in Uniform Region	+-250,000	nT
Uniform Field Dimension	0.4	m
Field Uniformity	2	%
Control Range	+-250,000	nT
Magnetic Field Noise	400	nT
Control Resolution	800 to 4,000	nT
Accuracy in Uniform Region	7	%
Angular Accuracy	5	°

6.4 HITL Test Results

A subset of the tests presented with the combined EKF in simulation are presented with added hardware noise, which shows the effect the hardware has on the estimation. The same initial conditions were used as for the combined EKF. In order to obtain long strings of data in a short period of time, the HITL tests were run at 10 times speed for 2 days of simulation time, or 5 hours of actual time. Separate tests were therefore needed to show the convergence and steady state error.

Due to the increased noise in the HITL measurements, the measurement noise matrix was increased to account for the extra sources of error. Table 6-2 outlines the different noise values employed for the different tests. The higher measurement covariance resulted in faster convergence and less initial bounce in the estimated parameters as the measurements were not followed as closely.

Table 6-2 Diagonal elements of the measurement noise matrix used for HITL testing.

Test conditions	Measurement covariance (standard deviation)
Simulated conditions	40000 (200 nT)
HITL measurement noise	1000000 (1000 nT)
HITL with Helmholtz cage	6250000 (2500 nT)

The convergence of the HITL simulations is shown in Table 6-3. For comparison, corresponding results from the simulated cases are also included. For the HITL measurement noise tests, the overall position estimate converged slowly with significant overshoot in the error. The increased convergence time was expected as the HITL tests include increased noise from the satellite bus that needed to be overcome. The convergence times were in agreement with those found in the work of Laneve (1997) and Abdelrahman (2011).

Table 6-3 Convergence times observed in the HITL tests.

Test Case	**Attitude convergence time (orbits)**	**Orbit convergence time (orbits)**
Simulated conditions	0.9	9
HITL measurement noise	0.8	10
HITL with Helmholtz cage	0.7	0.5

While the overall position converged quite quickly for the HITL tests with the Helmholtz cage, the individual orbital elements took much longer to reach steady state values. As an example, the semi-major axis is presented in Figure 6-8, which also shows the disparity between the covariance and the convergence of the estimated parameter. The covariance more closely followed the convergence of the total position error due to the EKF being an under-determined system.

As expected, the convergence time for the HITL simulation was longer than for the simulated results as there was more noise to overcome. The convergence time for the attitude portion of the EKF showed a slight decrease which was due to the lack of errors from external torques. The possible attitude error sources had been removed in order to align the air bearing table with the Helmholtz cage. This allowed a reliable baseline for the attitude to be obtained.

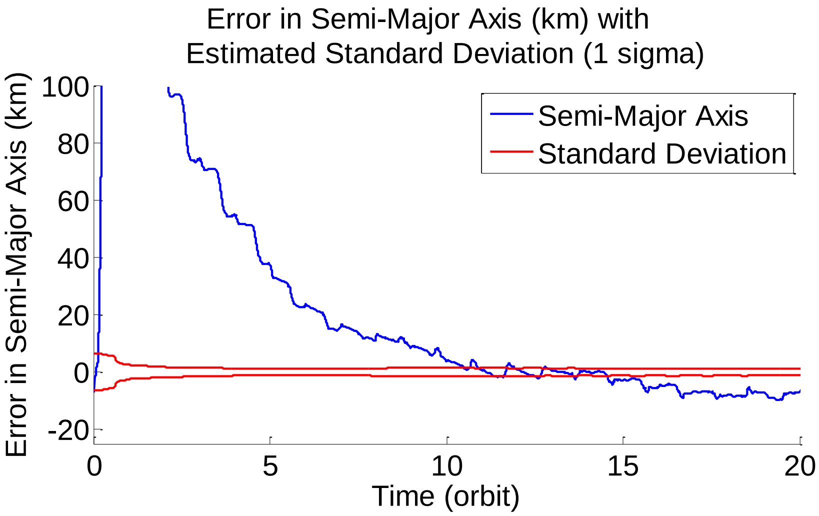

Figure 6-8 Typical convergence period for orbital elements and their standard deviations in tests involving the Helmholtz cage.

Convergence for the tests with the Helmholtz cage was much faster than the other tests. The shorter convergence period appeared to be due to the increased *R* matrix, which led to reduced bounce in the estimated parameters. The decrease in convergence time suggests that tuning may be able to improve the rate of convergence of the EKF in Chapter 5. The steady state orbit estimation statistics for the HITL simulations, due to the short test length, were taken from tests which started with an accurate initialization. The statistics are presented in Table 6-4 and Table 6-5. The statistics for the simulations are also included in the tables for comparison. Since the error sources due to attitude torques were not present in the HITL tests, a comparison between these tests and those reported previously is not valuable, and their statistics are omitted. Only the overall attitude error is shown in Table 6-6 as evidence that the added noise does not prevent the convergence of the EKF.

For the estimated orbital parameters in the HITL simulation, the standard deviation was similar to that of the simulated conditions. The semi-major axis and eccentricity were very similar to the previous simulations. The drag coefficient showed significant improvement over previous simulations, which may be due to the increased measurement noise covariance absorbing some of the modelling errors that showed up in the drag coefficient in the simulated tests. The remaining parameters showed significant increases in standard deviation. The mean anomaly and argument of perigee both showed similar increases as they are related parameters in circular orbits.

The tests with Helmholtz cage data showed order of magnitude increases in error due to the granularity of the Helmholtz cage output. The order of magnitude increase in the error was also present when looking at the error in estimated position and velocity. The dramatic increase in error shows that the Helmholtz cage was not precise enough to be used for orbit determination tests. The overall error in position and velocity for the HITL tests showed a three-fold increase in error over the simulated tests. This error was solely due to the added noise in the measurements from satellite bus sources.

Table 6-4 Steady state standard deviation of estimated orbit parameters in HITL tests.

Test Case	**Simulated conditions**	**HITL measurement noise**	**HITL with Helmholtz cage**
$d\hat{\alpha}$ **(Earth Radii)**	4.56×10^{-5}	4.12×10^{-5}	7.87×10^{-4}
$d\hat{\varepsilon}$	3.98×10^{-5}	3.93×10^{-5}	6.45×10^{-4}
$d\hat{\imath}$ **(rad)**	1.37×10^{-4}	2.46×10^{-4}	7.34×10^{-4}
$d\hat{\omega}$ **(rad)**	8.57×10^{-4}	1.22×10^{-3}	1.42×10^{-2}
$d\hat{\psi}$ **(rad)**	5.08×10^{-4}	1.61×10^{-3}	4.26×10^{-3}
$d\hat{m}$ **(rad)**	1.02×10^{-3}	1.57×10^{-3}	1.49×10^{-2}
$d\hat{b}^{\star}$	5.45×10^{-3}	3.41×10^{-3}	6.09×10^{-2}

As attitude filters have been more frequently investigated, and the HITL tests are focussed on validating the orbit portion of the EKF, the attitude results are shown here for completeness. As the air bearing table was aligned to the Helmholtz cage for the measurements presented in this chapter, and no rotation was applied to the HITL simulation, the identity quaternion, (0,0,0,1), was taken as the true attitude quaternion.

Table 6-5 Steady state 95th percentile of overall orbit errors in the HITL tests.

Test Case	**Simulated Conditions**	**HITL measurement noise**	**HITL with Helmholtz cage**
$\left\| d\hat{\dot{\vec{r}}} \right\|$ **(km/s)**	1.24×10^{-2}	3.54×10^{-2}	0.155
$\left\| d\hat{\vec{r}} \right\|$ **(km)**	12.3	35.7	175

The error in the total attitude knowledge showed a marked improvement in the HITL and Helmholtz cage tests, as was expected, due to the lack of disturbance torques in these tests. The Helmholtz cage test showed much more error than the HITL test, which was again due to the granularity of the cage output which can be seen in Figure 6-9. The granularity caused the magnetic field vector to point consistently in one direction then suddenly change to another direction. The magnetic field should have been changing smoothly from one to the other. The difference shows that the drive electronics for the Helmholtz cage gave poor results for attitude and orbit determination on the air bearing platform. Smaller granularity of the cage output settings would lead to a smoother magnetic field profile, and an improvement in performance of the algorithm on the air bearing table.

Table 6-6 Steady state 95th percentile of total attitude error in the HITL tests.

Test Case	**Simulated Conditions**	**HITL measurement noise**	**HITL with Helmholtz cage**
Attitude knowledge error (°)	9.22	0.861	6.75

Figure 6-9 shows a marked increase in granularity as the cage output increases. Helmholtz cage output in the z-axis was high for magnetic field values near and above 0 nT as the external magnetic field where the Helmholtz cage was located was strongly in the negative z direction. The Helmholtz cage needed to generate both the desired field as well as a compensating field to remove the external magnetic field. The x-axis showed similar behaviour, but with less external magnetic field to overcome.

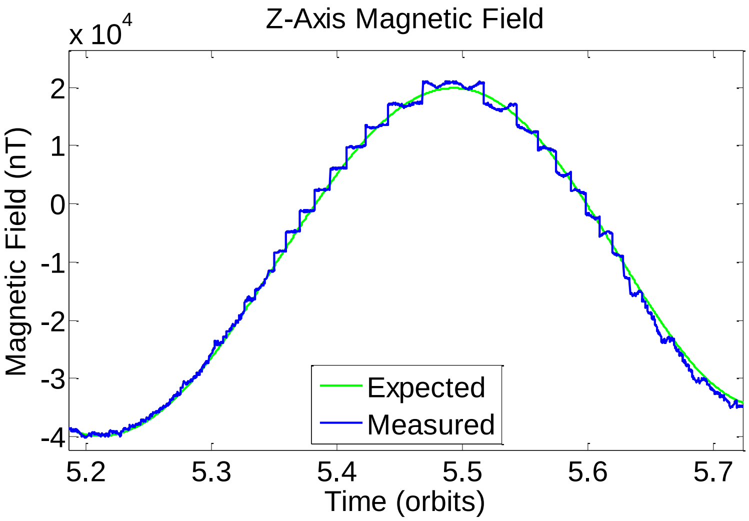

Figure 6-9 Comparison between measured magnetic field and desired magnetic field, showing granularity in cage output.

6.5 Summary of the HITL Simulations

Realistic hardware noise conditions, provided by a nanosatellite analog, were used to validate the results of the combined attitude and orbit determination EKF presented in Chapter 5. A Helmholtz cage was also evaluated for ground testing purposes. The validation tests used a nanosatellite analog to provide a realistic noise profile, while measurements were gathered. The nanosatellite analog included ACS sensors and actuators, an OBC, power system, and communications system. A calibration matrix was applied to the measurements, and the background magnetic field was removed. These measurements served as a noise profile to be added to the simulated measurement data. Convergence of the HITL tests took a similar amount of time as with the simulated tests, and faster with the Helmholtz cage, suggesting that more tuning is necessary to reduce the convergence times. The covariance convergence issues found in previous chapters persisted, and may require further tuning to remove. The HITL tests show that the simulations were optimistic, but reasonable. Several sources of error were not included in the HITL simulations. Specifically, solar panels can generate significant stray magnetic fields, but were not present on the satellite analog due to mounting and lighting considerations. In the HITL tests, the algorithm out performed NORAD TLEs after 10.5 days (170 orbits). The performance was adequate for many nanosatellite missions though the field of view for Earth pointing instruments would need to be larger than 5°, and the TLE update would need to be less frequent than once every 10.5 days for the algorithm to be useful. 35.7 km is still accurate enough for scheduling on a nanosatellite. Due to the discrepancy between the covariances and state vectors, the algorithm is best used as a backup to other systems, so that convergence can be determined ahead of time.

The Helmholtz cage consisted of three pairs of orthogonally mounted coils controlled by 12 V PWM filtered to provide consistent voltage between 0 and 12 V to the coils. The Helmholtz cage was able to provide realistic on orbit magnetic field values within 4,000 nT. The granularity of the Helmholtz cage output made it unsuitable for both attitude and orbit determination. In order to use the Helmholtz cage for ground tests of attitude and orbit determination algorithms, the granularity would need to be improved.

7 Conclusions and Future Work

While other attempts are being made to estimate attitude and orbit using magnetometers, they are limited to high accuracy sensors and multi sensor systems (Abdelrahman and Park, 2011; Laneve and Curti, 1997; Jung and Psiaki, 2002). This research has determined the applicability of using only magnetometers for both orbit and attitude determination on nanosatellites using low-grade COTS components. Separate Extended Kalman Filters (EKFs) have been presented for attitude determination and orbit determination. These filters served to set a baseline which was used to investigate the effects of running the two EKFs concurrently. Concurrent estimation of the attitude and orbit of a nanosatellite was simulated and validated using Hardware-In-The-Loop (HITL) testing. The testing served to show the effectiveness of the algorithm for a nanosatellite platform. HITL testing was also used to evaluate the effectiveness of a Helmholtz cage for future HITL attitude and orbit determination tests. This chapter summarizes the results of this research, and presents avenues for possible improvements and future study.

7.1 Conclusions

An MEKF was created to estimate the attitude quaternion and the angular rates of a nanosatellite. The attitude was propagated using fourth order Runge-Kutta numerical integration of the quaternion kinematics and Newton-Euler equations. The MEKF accomplished the estimation by comparing three-axis magnetic field measurements to the IGRF. Test cases showed that the MEKF created was able to handle the attitude determination needs of nanosatellites. The tests set the baseline for attitude determination to be used for comparison with combined orbit and attitude determination at a convergence time of 0.5 orbits and an accuracy of 5.5°. Convergence was more rapid than expected, while the accuracy was consistent with the work of others (Abdelrahman and Park, 2011).

An EKF was created to estimate the Brouwer mean orbital elements and drag term of a nanosatellite. The orbital elements were propagated using the secular variations portion of the SGP4 propagator. The EKF accomplished the estimation by comparing the magnitude of magnetic field measurements to the IGRF. Test cases showed that the EKF created was able to handle orbit determination for nanosatellites autonomously with magnetometers. The tests set the baseline for orbit determination to be used for comparison with combined orbit and attitude determination at a convergence time of 4 orbits and an accuracy of 27.9 km. The EKF converged more rapidly, and to greater accuracy than expected from the work of others (Abdelrahman and Park, 2011; Jung and Psiaki, 2002).

The two EKFs from Chapters 3 and 4 were combined by running both dynamic updates, then both measurement updates concurrently. To account for the effect the orbit had on the measurement uncertainty, an additional term, calculated from the orbit knowledge, was added to the measurement noise in the attitude determination portion of the combined EKF. The performance across the range of typical nanosatellite orbit conditions was examined. The effects magnetometer-based orbit determination has on attitude determination was shown to be a loss in accuracy of nearly 4°. This loss in accuracy was consistent with what was expected from the work of others (Abdelrahman and Park, 2011; Psiaki et al., 1990). As expected, combining the EKFs had no effect on the orbit determination. Convergence stayed largely the same, with a slight delay in attitude convergence due to a lack of orbit knowledge.

In evaluating the applicability of the algorithm to nanosatellite missions in general, several mission requirements were determined. The 9° attitude knowledge accuracy may make it difficult to use the algorithm for active attitude control. The accuracy of the orbit determination makes this algorithm useful in any situation where TLEs are likely to be updated less than once a week.

Realistic hardware noise conditions, provided by a nanosatellite analog, were used to validate the results of the combined attitude and orbit determination EKF. A Helmholtz cage was also evaluated for ground testing purposes. The validation tests used a nanosatellite analog to provide a realistic noise profile while measurements were gathered. The nanosatellite analog included ACS sensors and actuators, an OBC, power system, and communications system. A calibration matrix was applied to the measurements, and the background magnetic field was removed. These measurements served as a noise profile to be added to the simulated measurement data. Convergence of the HITL tests took a similar amount of time as with the simulated tests. The HITL tests showed that the simulations were optimistic, but reasonable. In the HITL tests, the orbit accuracy was shown to be 35.7 km, sufficient for the algorithm to out perform NORAD TLEs after 10.5 days (170 orbits). This accuracy was consistent with what was expected extrapolating from the work of others (Abdelrahman and Park, 2011; Jung and Psiaki, 2002).

The Helmholtz cage consisted of three pairs of orthogonally mounted coils controlled by filtered 12 V PWM. The Helmholtz cage was able to provide realistic on orbit magnetic field values within 4,000 nT. The granularity of the Helmholtz cage output made it unsuitable for both attitude and orbit determination. To use the Helmholtz cage for ground tests of attitude and orbit determination algorithms, the granularity would need to be improved.

One problem encountered across all the EKFs was a difficulty identifying convergence, as the covariances did not reflect the accuracy of the estimates during the convergence period. Due to this discrepancy, it is suggested that the algorithms presented here be used primarily as backup systems. Alternatively, the EKFs can be initialized with accurate estimates through the use of additional attitude sensors and TLEs. Accuracy of better than 35.7 km is sufficient for most nanosatellite operations. Fine knowledge of measurement locations would be better done with GPS or frequent updates of TLEs, if the mission allows. Autonomous scheduling can be facilitated by the algorithm presented.

7.2 Future Considerations

In order to improve the results presented here, several avenues bear investigating. The attitude results can be improved by including the disturbance torques in the dynamic model. To improve both the convergence times and the covariances during convergence, additional tuning of the process noise matrix, and additional modeling of non-Gaussian process and measurement noises to satisfy the assumptions of the EKF can be investigated. One possible way to identify the convergence of the EKFs may be to use a smoothing filter as comparison. The next steps for HITL tests are to include the remaining nanosatellite bus sources of magnetic field, including solar panels, and testing using a Helmholtz cage with small granularity in the generated magnetic fields to test the algorithm in conditions closer to those encountered on orbit. To fully demonstrate the functionality of the algorithm on nanosatellite platforms, actual, on orbit nanosatellite data should be used. In order to use the Helmholtz cage for ground tests of attitude and orbit determination algorithms, the granularity must be improved by increasing the number of PWM setpoints. Another possible avenue for improving the algorithm is to include other sensors that are frequently included on nanosatellite platforms, such as Sun sensors, similar to the investigations of Psiaki (2002) for larger satellites.

References

Abdelrahman, M. and S. Park. 2011. Simultaneous Spacecraft Attitude and Orbit Estimation using Magnetic Field Vector Measurements. *Aerospace Science and Technology*, Vol. 15, pp. 653-669.

Acuña, M. H. 2002. Space-Based Magnetometers. *Review of Scientific Instruments*, Vol. 73, No. 11, pp. 3717-3736.

Analog Devices Inc. 2011. High Performance, Digital Output Gyroscope ADXRS453 datasheet. *http://www.analog.com/static/imported-files/data_sheets/ADXRS453.pdf* Accessed: Mar., 7 2014.

Atmospheric and Environmental Research Inc. 2014. SEET: Space Environment and Effects Tool for AGI's Systems Tool Kit (STK). *http://www.agi.com/resources/help/online/stk/STK/pdf/SEET_manual.pdf* Accessed: July 30 2014.

Blue Canyon Technologies. 2013. Nano Star Tracker Datasheet. *http://bluecanyontech.com/wp-content/uploads/2013/06/BCT-Nano-Star-Tracker-datasheet-1.1.pdf* Accessed: June, 4 2014.

Borschiov, K. 2012. Generic on-Board-Computer Hardware and Software Development for Nanosatellite Applications. M.Sc. thesis, York University, 112p.

Bouwmeester, J. and J. Guo. 2010. Survey of Worldwide Pico-and Nanosatellite Missions, Distributions and Subsystem Technology. *Acta Astronautica*, Vol. 67, No. 7, pp. 854–862.

California Polytechnic State University. 2014. CubeSat Design Specification Rev. 13. *www.cubesat.org/images/developers/cds_rev13_final.pdf* Accessed: June 3 2014.

Cannata, M. N. 2010. Development of a Sun Vector Determination Algorithm for CubeSat-Class Spacecraft. M.Sc. thesis, York University, 144p.

Caruso, M. J., C. H. Smith, T. Bratland, and R. Schneider. 1998. A New Perspective on Magnetic Field Sensing. *Proceedings of Sensors Expo*. Oct., pp. 195-213.

Community Coordinated Modeling Center. 2011. About Magnetospheric Field Models at CCMC. *http://ccmc.gsfc.nasa.gov/modelweb/magnetos/about_magnetos1.html* Accessed: June 8 2014.

Crassidis, J. L., F. L. Markley, and Y. Cheng. 2007. A Survey of Nonlinear Attitude Estimation Methods. *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 1, pp. 12-28.

Digi-Key Corporation. 2012. Magnetic Sensors - Hall Effect, Digital Switch, Linear, Compass (ICs). http://www.digikey.ca/product-search/en/sensors-transducers/magnetic-sensors-hall-effect-digital-switch-linear-compass-ics/1967232 Accessed: Aug. 30 2012.

Fat Quarters Software and Electronics. 2012. FGM - series Magnetic Field Sensors. http://www.fatquarterssoftware.com/uploads/2/6/6/8/2668219/fgm_ds.pdf Accessed: Aug. 30 2012.

Flury, W. 1995. The Space Debris Environment of the Earth. *Earth, Moon, and Planets*, Vol. 70, No. 1-3, pp. 79-91.

Friis-Christensen, E., H. Luhr, and G. Hulot. 2006. Swarm: A Constellation to Study the Earth's Magnetic Field. *Earth, Planets, and Space*, Vol. 58, No. 4, pp. 351-358.

Glatzmaier, G. A. 2010. The Geodynamo. http://www.es.ucsc.edu/~glatz/geodynamo.html Accessed: July 12 2012.

Greene, M. and R. Zee. 2009. Increasing the Accuracy of Orbital Position Information from NORAD SGP4 using Intermittent GPS Readings. *Proceedings of AIAA/USU 23rd Annual Conference on Small Satellites*, Aug. 13, pp. 1-6.

Gregory, B. S. 2001. Attitude Control System Design for ION, the Illinois Observing Nanosatellite. M.Sc. thesis, University of Illinois, 105p.

Haque, S. E., G. Teel, M. Keidar, E. F. Agasid, O. T. Gazulla, A. D. Perez, G. T. Trinh, and E. A. Uribe. 2013. Micro Cathode Arc Thruster PhoneSat Experiment for Small Satellites. *Proceedings of the 33rd International Electric Propulsion Conference*, Washington , D.C., Oct. 6–10, pp. 1-10.

Honeywell International Inc. 2011. 3-Axis Digital Compass IC HMC5883L. http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/Magneto/HMC5883L-FDS.pdf Accessed: Aug. 30 2012.

Hoots, F. R., P. W. Schumacher Jr., and R. A. Glover. 2004. History of Analytical Orbit Modeling in the U.S. Space Surveillance System. *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 2, pp. 174-185.

Innovative Solutions in Space. 2014a. CubeSat Sun Sensor. http://www.cubesatshop.com/index.php?page=shop.product_details&product_id=104&flypage=flypage.tpl&pop=0&option=com_virtuemart&Itemid=65&vmcchk=1&Itemid=65 Accessed: July 30 2014.

Innovative Solutions in Space. 2014b. CubeSense. http://www.cubesatshop.com/index.php?page=shop.product_details&flypage=flypage.tpl&product_id=107&category_id=7&option=com_virtuemart&Itemid=69&vmcchk=1&Itemid=69 Accessed: July 30 2014.

Jung, H. and M. L. Psiaki. 2002. Tests of Magnetometer/Sun-Sensor Orbit Determination using Flight Data. *Journal of Guidance, Control, and Dynamics,* Vol. 25, No. 3, pp. 582-590.

Kahr, E., S. Skone, and K. O'Keefe. 2010. Orbit Determination for the CanX-2 Nanosatellite using Intermittent GPS Data. *Proceedings of ION-GNSS-2010,* Manassas, Virginia, Sept. 21-24, pp. 2117-2125.

Kelso, T. S. 2014. NORAD Two-Line Element Sets Historical Archives. *http://celestrak.com/NORAD/archives/request.asp* Accessed: Mar 12 2014.

Kelso, T. S. 2007. Validation of SGP4 and IS-GPS-200D Against GPS Precision Ephemerides. *Proceedings of the 17th AAS/AIAA Space Flight Mechanics Conference,* Sedona, Arizona, Jan. 28 - Feb. 1, pp. 1-14.

Kianfar, K., A. J. Mahalleh, and A. Moridi. 2011. Select the Appropriate Model for the Earth's Magnetic Field. *Proceedings of the 5th European Symposium on Computer Modeling and Simulation,* Madrid, Spain, Nov. 16-18, pp. 275-280.

Kyung Hee University. 2014. Kyung Hee University Launches Miniature Satellites. *http://khu.ac.kr/eng/about/news_view.jsp?idx=228&iPage=1* Accessed: June 8 2014.

Laneve, G. and F. Curti. 1997. An Orbit Determination Approach for Small Satellites. *Proceedings of the 12th International Symposium on Space Flight Dynamics,* European Space Operation Centre, Darmstacht, Germany, Jun 2-6, pp. 275-282.

Larson, W. J., and J. R. Wertz, eds. 1999. *Space Mission Analysis and Design*, 3rd ed., Microcosm Press and Kluwer Academic Publishers.

Lee, R., S. Bisnath, P. Gavigan, and A. Ng. 2012. Small is Beautiful - 1st Report from the Canadian Nanosatellite Workshop. *1st Canadian Nanosatellite Workshop,* Quebec City, Canada, Apr. 23, pp. 1-25.

Lenz, J. and A. S. Edelstein. 2006. Magnetic Sensors and their Applications. *IEEE Sensors Journal,* Vol. 6, No. 3, pp. 631-649.

Li, J., M. Post, T. Wright, and R. Lee. 2013. Design of Attitude Control Systems for Cubesat-Class Nanosatellite. *Journal of Control Science and Engineering,* Vol. 2013, pp. 4-18.

Lin, R. P., J. G. Sample, T. J. Immel, D. Lee, T. S. Horbury, H. Jin, J. Seon, L. Wang, E. C. Roelof, E. Lee, G. K. Parks, and H. Vo. 2012. Stereo ENA Imaging of the Ring Current and Multi-Point Measurements of Suprathermal Particles and Magnetic Fields by TRIO-CINEMA. *AGU Fall Meeting Abstracts,* Vol. 1, pp. 2178.

Long, M., A. Lorenz, G. Rodgers, E. Tapio, G. Tran, K. Jackson, and R. Twiggs. 2002. A Cubesat Derived Design for Unique Academic Research Mission in Earthquake Signature

Detection. *proceedings of the 16th Annual AIAA/USU Conference on Small Satellites,* Logan, Utah, Aug. 12-15, pp. 1-17.

Malin, S. 1987. *Historical Introduction to Geomagnetism,* Academic Press.

Markley, F. L. 2003. Attitude Error Representations for Kalman Filtering. *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 2, pp. 311-317.

Maus, S., S. Macmillan, S. McLean, B. Hamilton, A. Thomson, M. Nair, and C. Rollins. 2009. The US/UK World Magnetic Model for 2010-2015. *ftp://ftp.ngdc.noaa.gov/ionosonde/request/robr/WMM2010_TR_preliminary.pdf* Accessed: Aug. 30 2012.

Mehrjardi, F. M. and M. Mirshams. 2010. Design and Manufacturing of a Research Magnetic Torquerer Rod. *Contemporary Engineering Sciences,* Vol. 3, No. 5, pp. 227-236.

Merayo, J. M. G., P. Brauer, F. Primdahl, J. R. Petersen, and O. V. Nielsen. 2000. Scalar Calibration of Vector Magnetometers. *Measurement Science and Technology*, Vol. 11, pp. 120-132.

Merrill, R. T., and M. W. McElhinny. 1982. *The Earth's Magnetic Field,* Academic Press.

Mobley, F. F., L. D. Eckard, G. H. Fountain, and G. W. Ousley. 1980. Magsat - A New Satellite to Survey the Earth's Magnetic Field. *IEEE Transactions on Magnetics*, Vol. 16, No. 5, pp. 758-760.

Montenbruck, O., and E. Gill. 2005. *Satellite Orbits,* Springer-Verlag Berlin Heidelberg.

Nair, M. 2014a. Geomagnetic Field Models at NGDC. *http://www.ngdc.noaa.gov/geomag/models.shtml* Accessed: Aug. 27 2014.

Nair, M. 2014b. International Geomagnetic Reference Field. *http://www.ngdc.noaa.gov/IAGA/vmod/igrf.html* Accessed: Aug. 1 2014.

Ochoa, B. and S. Belongie. 2006. Covariance Propagation for Guided Matching. *http://vision.ucsd.edu/sites/default/files/ochoa06.pdf* Accessed: Aug. 28 2014.

Olson, W. P. and K. A. Pfitzer. 1974. A Quantitative Model of the Magnetospheric Magnetic Field. *Journal of Geophysical Research,* Vol. 79, No. 25, pp. 3739-3748.

Pfitzer, K. A. and W. P. Olson. 2011. Olson-Pfitzer Field Model 1974. *http://ccmc.gsfc.nasa.gov/modelweb/magnetos/olson.html* Accessed: Aug. 30 2012.

Pisacane, Vincent L., and Robert Clark Moore, eds. 1994. *Fundamentals of Space Systems,* Oxford University Press.

Post, M., J. Li, T. Wright, H. Hakima, and R. Lee. 2013. Design and Testing of a Magnetic Field Simulator for Nanosatellite Air Bearing System. Unpublished manuscript, York University, Toronto, Canada.

Psiaki, M. 1999. Autonomous LEO Orbit Determination from Magnetometer and Sun Sensor Data. *Journal of Guidance, Control, and Dynamics,* Vol. 22, No. 2, pp. 296-304.

Psiaki, M. L., F. Martel, and P. K. Pal. 1990. Three-Axis Attitude Determination Via Kalman Filtering of Magnetometer Data. *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 3, pp. 506-514.

Roberts, P. H. and G. A. Glatzmaier. 2000. Geodynamo Theory and Simulations. *Reviews of Modern Physics,* Vol. 72, No. 4, pp. 1081-1123.

Sarda, K., C. Grant, S. Eagleson, D. D. Kekez, A. Shah, and R. E. Zee. 2009. Canadian Advanced Nanospace Experiment 2 Orbit Operations: One Year of Pushing the Nanosatellite Performance Envelope. *Proceedings of the 23rd Annual AIAA/USU Conference on Small Satellites,* Aug.13, pp. 1-46.

Schwindt, P. D. D., S. Knappe, V. Shah, L. Hollberg, and J. Kitching. 2004. Chip-Scale Atomic Magnetometer. *Applied Physics Letters,* Vol. 85, No. 26, pp. 6409-6411.

Silani, E. and M. Lovera. 2005. Magnetic Spacecraft Attitude Control: A Survey and some New Results. *Control Engineering Practice,* Vol. 13, No. 3, pp. 357-371.

Stern, D. P. 1989. A Brief History of Magnetospheric Physics before the Spaceflight Era. *Reviews of Geophysics,* Vol. 27, No. 1, pp. 103-114.

Süli, E., and D. F. Mayers. 2003. *An Introduction to Numerical Analysis,* Cambridge University Press.

UTIAS SFL. 2014. CanX-2: Mission Objectives. *http://utias-sfl.net/?page_id=274* Accessed: July 30 2014.

Ustrzycki, T. 2011. Spherical Air Bearing Testbed for Nanosatellite Attitude Control Development. M.Sc. Thesis, York University, 107p.

Vallado, D. A., P. Crawford, R. Hujsak, and T. S. Kelso. 2006. Revisiting Spacetrack Report #3. *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference,* Keystone, Colorado, Aug. 21–24, pp. 1-88.

Welch, G. and G. Bishop. 2006. An Introduction to the Kalman Filter. *http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf* Accessed: Feb. 8 2014.

Wertz, J. R., ed. 1978. *Spacecraft Attitude Determination and Control,* D. Reidel Publishing Company.

Winch, D. E., D. J. Ivers, J. P. R. Turner, and R. J. Stening. 2004. Geomagnetism and Schmidt Quasi-Normalization. *Geophysics Journal International*, Vol. 160, No. 2, pp. 487-504.

Appendix A CubeSat™ Orbit Survey

The following table, detailing the initial orbits used by many Cubesats™, was compiled for this research using the first available NORAD TLEs retrieved from CelesTrak (Kelso, 2014) for satellites under 10 kg.

Satellite Name	NORAD ID	Altitude of Apogee (km)	Inclination (°)	Eccentricity
DTUsat	27842	821	98.7	0.0008
CUTE-I	27844	823	98.7	0.0008
QuakeSat	27845	825	98.7	0.0008
AAU Cubesat	27846	821	98.7	0.0012
CanX-1	27847	823	98.7	0.0017
CubeSat XI-IV	27848	825	98.8	0.0005
UWE-1	28892	692	98.2	0.0016
CubeSat XI-V	28895	692	98.2	0.0019
CUTE 1.7 + APD	28941	502	98.2	0.0300
GeneSat-1	29655	419	40.0	0.0005
CSTB1	31122	707	98.1	0.0084
MAST	31126	712	98.1	0.0093
Libertad-1	31128	701	98.1	0.0086
CP-3	31129	717	98.1	0.0103
CAPE1	31130	717	98.1	0.0102
CanX-6	32784	624	98.0	0.0016
CUTE 1.7 + APD II	32785	625	98.0	0.0013
Compass One	32787	639	97.9	0.0033
AAUsat-II	32788	622	98.0	0.0016
Delfi-C3	32789	622	98.0	0.0015
CanX-2	32790	621	98.0	0.0014
SEEDS (2)	32791	623	98.0	0.0015
PRISM	33493	646	98.0	0.0019
STARS	33498	652	98.0	0.0016
KKS-1	33499	656	98.0	0.0011
PharmaSat	35002	449	40.5	0.0027
CP-6	35003	449	40.5	0.0030
HawkSat I	35004	449	40.5	0.0029
AeroCube 3	35005	449	40.5	0.0028

Satellite Name	NORAD ID	Altitude of Apogee (km)	Inclination (°)	Eccentricity
SwissCube	35932	716	98.3	0.0007
BeeSat	35933	714	98.3	0.0005
UWE-2	35934	715	98.3	0.0006
İTÜpSAT1	35935	717	98.3	0.0007
Waseda-SAT2	36574	302	30.0	0.0011
Negai	36575	300	30.0	0.0013
Studsat	36796	627	98.2	0.0011
TIsat 1	36799	625	98.1	0.0016
QbX2	37245	301	34.5	0.0022
SMDC-ONE	37246	299	34.5	0.0021
Perseus 003	37247	298	34.5	0.0021
Perseus 001	37248	299	34.5	0.0022
QbX1	37249	299	34.5	0.0020
Perseus 002	37250	301	34.5	0.0022
Perseus 000	37251	283	34.5	0.0041
DICE-F	37851	634	101.7	0.0254
DICE-Y	37852	635	101.7	0.0256
RAX-2	37853	635	101.7	0.0255
AubieSat-1	37854	634	101.7	0.0256
M-Cubed	37855	634	101.7	0.0258
e-st@r	38079	873	69.5	0.0781
Goliat	38080	874	69.5	0.0780
MaSat-1	38081	874	69.5	0.0780
XaTcobeo	38082	874	69.5	0.0781
PW-Sat	38083	874	69.5	0.0780
ROBUSTA	38084	874	69.5	0.0781
UniCubeSat GG	38085	874	69.5	0.0782
AAUSAT3	39087	777	98.6	0.0010
BeeSat-3	39134	566	64.9	0.0019
SOMP	39135	566	64.9	0.0019
BeeSat-2	39136	566	64.9	0.0017
Bell	39142	248	51.6	0.0016
Alexander	39144	222	51.6	0.0017
Graham	39146	221	51.6	0.0010
NEE-01 Pegasus	39151	640	98.1	0.0017
CUBEBUG 1	39153	639	98.1	0.0017
ESTCube-1	39161	662	98.1	0.0009

Appendix B Partial Derivatives of the Magnetic field

The partial derivatives for the magnetic field in the ECEF reference frame with respect to the position in the ECEF reference frame are based on those from (Montenbruck and Gill, 2005), but converted to use Schmidt quasi-normalized associated Legendre functions through the use of equations (2-18) through (2-23). These partial derivatives are as follows:

$$\frac{d\vec{b}_x}{dx} = \sum -\frac{R_E^{n+2}}{r^{n+3}}\frac{1}{2}\left\{\sqrt{\frac{(n+4)!}{n!\,2}}\, g_n^m v_{n+2}^{m+2} - \frac{(n+2)!}{n!}\, g_n^m v_{n+2}^m\right\}, m = 0 \tag{B-1}$$

$$\begin{aligned}\frac{d\vec{b}_x}{dx} = \sum -\frac{R_E^{n+2}}{r^{n+3}}\frac{1}{4}\Bigg\{&\sqrt{\frac{(n+m+4)!}{(n+m)!}}\left[g_n^m v_{n+2}^{m+2} + h_n^m w_{n+2}^{m+2}\right] \\ &- \frac{(n-m+2)!}{(n-m)!}\sqrt{\frac{(n+m+2)!\,(n-m)!}{(n-m+2)!\,(n+m)!}}\left[3 g_n^m v_{n+2}^m\right. \\ &\left.+ h_n^m w_{n+2}^m\right]\Bigg\}, m = 1\end{aligned} \tag{B-2}$$

$$\begin{aligned}\frac{d\vec{b}_x}{dx} = \sum -\frac{R_E^{n+2}}{r^{n+3}}\frac{1}{4}\Bigg\{&\sqrt{\frac{(n+m+4)!}{(n+m)!}}\left[g_n^m v_{n+2}^{m+2} + h_n^m w_{n+2}^{m+2}\right] \\ &+ \sqrt{\frac{2(n-m+4)!}{(2-\delta_m^2)(n-m)!}}\left[g_n^m v_{n+2}^{m-2} + h_n^m w_{n+2}^{m-2}\right] \\ &- \frac{2(n-m+2)!}{(n-m)!}\sqrt{\frac{(n+m+2)!\,(n-m)!}{(n-m+2)!\,(n+m)!}}\left[g_n^m v_{n+2}^m\right. \\ &\left.+ h_n^m w_{n+2}^m\right]\Bigg\}, m \geq 2\end{aligned} \tag{B-3}$$

$$\frac{d\vec{b}_x}{dy} = \sum -\frac{R_E^{n+2}}{r^{n+3}}\frac{1}{2}\sqrt{\frac{(n+4)!}{n!\,2}}\,g_n^m w_{n+2}^{m+2}, m = 0 \qquad \text{(B-4)}$$

$$\begin{aligned}\frac{d\vec{b}_x}{dy} = \sum -\frac{R_E^{n+2}}{r^{n+3}}\frac{1}{4}\Bigg\{&\sqrt{\frac{(n+m+4)!}{(n+m)!}}\,[g_n^m w_{n+2}^{m+2} - h_n^m v_{n+2}^{m+2}] \\ &- \frac{(n-m+2)!}{(n-m)!}\sqrt{\frac{(n+m+2)!\,(n-m)!}{(n-m+2)!\,(n+m)!}}\,[g_n^m w_{n+2}^m \\ &+ h_n^m v_{n+2}^m]\Bigg\}, m = 1\end{aligned} \qquad \text{(B-5)}$$

$$\begin{aligned}\frac{d\vec{b}_x}{dy} = \sum -\frac{R_E^{n+2}}{r^{n+3}}\frac{1}{4}\Bigg\{&\sqrt{\frac{(n+m+4)!}{(n+m)!}}\,[g_n^m w_{n+2}^{m+2} - h_n^m v_{n+2}^{m+2}] \\ &+ \sqrt{\frac{2(n-m+4)!}{(2-\delta_m^2)(n-m)!}}\,[h_n^m v_{n+2}^{m-2} - g_n^m w_{n+2}^{m-2}]\Bigg\}, m \geq 2\end{aligned} \qquad \text{(B-6)}$$

$$\frac{d\vec{b}_x}{dz} = \sum -\frac{R_E^{n+2}}{r^{n+3}}(n+1)\sqrt{\frac{(n+3)!}{(n+1)!\,2}}\,g_n^m v_{n+2}^{m+1}, m = 0 \qquad \text{(B-7)}$$

$$\begin{aligned}\frac{d\vec{b}_x}{dz} = \sum -\frac{R_E^{n+2}}{r^{n+3}}\frac{1}{2}\Bigg\{&(n-m+1)\sqrt{\frac{(n+m+3)!}{(n+m)!\,(n-m+1)}}\,[g_n^m v_{n+2}^{m+1} \\ &+ h_n^m w_{n+2}^{m+1}] \\ &- \frac{(n-m+3)!}{(n-m)!}\sqrt{\frac{2(n+m+1)(n-m)!}{(2-\delta_m^1)(n-m+3)!}}\,[g_n^m v_{n+2}^{m-1} \\ &+ h_n^m w_{n+2}^{m-1}]\Bigg\}, m > 0\end{aligned} \qquad \text{(B-8)}$$

$$\frac{d\vec{b}_y}{dz} = \sum -\frac{R_E^{n+2}}{r^{n+3}}(n+1)\sqrt{\frac{(n+3)!}{(n+1)!\,2}}\,g_n^m w_{n+2}^{m+1}, m = 0 \qquad \text{(B-9)}$$

$$
\begin{aligned}
\frac{d\vec{b}_y}{dz} = \sum -\frac{R_E^{n+2}}{r^{n+3}}\frac{1}{2}\Bigg\{ & (n-m+1)\sqrt{\frac{(n+m+3)!}{(n+m)!\,(n-m+1)}}\,[g_n^m w_{n+2}^{m+1} \\
& - h_n^m v_{n+2}^{m+1}] \\
& + \frac{(n-m+3)!}{(n-m)!}\sqrt{\frac{2(n+m+1)(n-m)!}{(2-\delta_m^1)(n-m+3)!}}\,[g_n^m w_{n+2}^{m-1} \\
& - h_n^m v_{n+2}^{m-1}]\Bigg\}, m > 0
\end{aligned}
\tag{B-10}
$$

$$
\frac{d\vec{b}_z}{dz} = \sum -\frac{R_E^{n+2}}{r^{n+3}}\sqrt{\frac{(n+m+2)!\,(n-m+2)!}{(n+m)!\,(n-m)!}}\,[g_n^m v_{n+2}^m + h_n^m w_{n+2}^m] \tag{B-11}
$$

$$
\frac{d\vec{b}_y}{dy} = -\frac{d\vec{b}_x}{dx} - \frac{d\vec{b}_z}{dz} \tag{B-12}
$$

$$
\frac{d\vec{b}_y}{dx} = \frac{d\vec{b}_x}{dy} \tag{B-13}
$$

$$
\frac{d\vec{b}_z}{dx} = \frac{d\vec{b}_x}{dz} \tag{B-14}
$$

$$
\frac{db_z}{dy} = \frac{db_y}{dz} \tag{B-15}
$$

Appendix C SGP4 Propagator and Partial Derivatives

The dynamic and measurement models for orbit determination are based on (Beck, 2012) and require some of the included functions. The partial derivatives are calculated, using the chain rule, alongside the calculations used in the propagator as shown in the following code:

Secular Variations

```
function [y,dy]=sgp4svd(x,t)
% performs secular update of the brouwer mean orbital elements using the near
earth portion of the sgp4 propagator
%
% ---------------------------------------------------------------------
% Author:
%   Thomas Wright
% original matlab code:
%   Jeff Beck
%   beckja@alumni.lehigh.edu
%    current :
%               7 may 08  david vallado
%                           update small eccentricity check
%    changes :
%              16 nov 07  david vallado
%                           misc fixes for better compliance
%   1.0 (aug 7, 2006) - update for paper dav
% original Vallado C++ version:
%   author        : david vallado          719-573-2600   28 jun 2005
%   references    :
%     hoots, roehrich, norad spacetrack report #3 1980
%     hoots, norad spacetrack report #6 1986
%     hoots, schumacher and glover 2004
%     vallado, crawford, hujsak, kelso  2006
% ---------------------------------------------------------------------
% INPUTS:
%   x   a 7 by 1 array of Brouwer mean orbital elements
%       (1) ao      Earth radii     mean semi-major axis
%       (2) ecco    unitless        mean eccentricity
%       (3) inclo   radians         mean inclination
%       (4) argpo   radians         mean argument of perigee
%       (5) nodeo   radians         mean RAAN
%       (6) mo      radians         mean mean anomaly
%       (7) bstar   1/Earth radius  drag coefficient
%   t   minutes     the time since the epoch of the input elements
%
% OUTPUTS:
%   y   a 7 by 1 array of mean orbital elements at the new epoch
%   dy  a 7 by 7 matrix of the Jacobian of the SGP4 secular variation
%
```

```
%% initialize necesary constants

%  // sgp4fix identify constants and allow alternate values
global tumin mu radiusearthkm xke j2 j3 j4 j3oj2
vkmpersec     = radiusearthkm * xke/60.0;

twopi = 2.0 * pi;
ss     = 78.0 / radiusearthkm + 1.0;
qzms2t = ((120.0 - 78.0) / radiusearthkm)^4;
x2o3  = 2.0 / 3.0;
% sgp4fix divisor for divide by zero check on inclination
% the old check used 1.0 + cos(pi-1.0e-9), but then compared it to
% 1.5 e-12, so the threshold was changed to 1.5e-12 for consistancy
temp4    =   1.5e-12;

%% initialize mean elements and their partial derivatives
ao      = x(1);
daodx=[1,0,0,0,0,0,0]; % partial derivative of ao
no      = xke/(ao^1.5);
dnodx=-1.5*xke*daodx/(ao^2.5); % partial derivative of no
ecco    = x(2);
deccodx=[0,1,0,0,0,0,0]; % partial derivative of ecco
inclo   = x(3);
dinclodx=[0,0,1,0,0,0,0]; % partial derivative of inclo
argpo   = x(4);
dargpodx=[0,0,0,1,0,0,0]; % partial derivative of argpo
nodeo   = x(5);
dnodeodx=[0,0,0,0,1,0,0]; % partial derivative of nodeo
mo      = x(6);
dmodx=[0,0,0,0,0,1,0]; % partial derivative of mo
bstar   = x(7);
dbstardx=[0,0,0,0,0,0,1]; % partial derivative of bstar

%% initialize zero quantities

method = 'n';   % 'n' implies near earth, 'd' implies deep space

%% initl

% /* ------------- calculate auxillary epoch quantities and their partial
derivatives ---------- */
eccsq  = ecco * ecco;
deccsqdx=2*ecco*deccodx; % partial derivative of eccsq
omeosq = 1.0 - eccsq;
domeosqdx=-deccsqdx; % partial derivative of omeosq
rteosq = sqrt(omeosq);
drteosqdx=0.5*domeosqdx/sqrt(omeosq); % partial derivative of rteosq
cosio  = cos(inclo);
dcosiodx=-sin(inclo)*dinclodx; % partial derivative of cosio
cosio2 = cosio * cosio;
dcosio2dx=2*cosio*dcosiodx; % partial derivative of cosio2

sinio = sin(inclo);
dsiniodx=cos(inclo)*dinclodx; % partial derivative of sinio
po    = ao * omeosq;
```

```
dpodx=daodx*omeosq+ao*domeosqdx; % partial derivative of po
con42 = 1.0 - 5.0 * cosio2;
dcon42dx=-5.0*dcosio2dx; % partial derivative of con42
con41 = -con42-cosio2-cosio2;
dcon41dx=-dcon42dx-dcosio2dx-dcosio2dx; % partial derivative of con41
posq  = po * po;    %
dposqdx=2*po*dpodx; % partial derivative of posq
rp    = ao * (1.0 - ecco);  %
drpdx=daodx*(1.0-ecco)+ao*(-deccodx); % partial derivative of rp

%% initialization of SGP4 quantities

   if ((omeosq >= 0.0 ) | ( no >= 0.0))
       isimp = 0;
       if (rp < (220.0 / radiusearthkm + 1.0))
           isimp = 1;
       end
       sfour  = ss; %
       dsfourdx=[0,0,0,0,0,0,0]; % partial derivative of sfour
       qzms24 = qzms2t;
       dqzms24dx=[0,0,0,0,0,0,0]; % partial derivative of qzms24
       perige = (rp - 1.0) * radiusearthkm; %
       dperigedx=(drpdx)*radiusearthkm; % partial derivative of perigee

       % /* - for perigees below 156 km, s and qoms2t are altered - */
       if (perige < 156.0)
           sfour = perige - 78.0;   %
           dsfourdx=dperigedx; % partial derivative of sfour
           if (perige < 98.0)
               sfour = 20.0;    %
               dsfourdx=[0,0,0,0,0,0,0]; % partial derivative of sfour
           end
           qzms24 = ((120.0 - sfour) / radiusearthkm)^4.0;
           dqzms24dx=4.0*(((120.0-sfour)/radiusearthkm)^3.0)* ...
               ((-dsfourdx)/radiusearthkm); % partial derivative of qzms24
           sfour  = sfour / radiusearthkm + 1.0;    %
           dsfourdx=dsfourdx/radiusearthkm; % partial derivative of sfour
       end
       pinvsq = 1.0 / posq; %
       dpinvsqdx=-dposqdx/(posq^2); % partial derivative of pinvsq

       tsi  = 1.0 / (ao - sfour);   %
       dtsidx=-(daodx-dsfourdx)/((ao-sfour)^2); % partial derivative of tsi
       eta  = ao * ecco * tsi;  %*
       detadx=daodx*ecco*tsi+ao*deccodx*tsi+ao*ecco*dtsidx; % partial
derivative of eta
       etasq = eta * eta;   %
       detasqdx=2*eta*detadx; % partial derivative of etasq
       eeta  = ecco * eta;  %
       deetadx=deccodx*eta+ecco*detadx; % partial derivative of eeta
       psisq = abs(1.0 - etasq);    %
       dpsisqdx=sign(1.0-etasq)*(-detasqdx); % partial derivative of psisq
       coef  = qzms24 * tsi^4.0;    %
       dcoefdx=(dqzms24dx*tsi^4.0)+4.0*(qzms24*dtsidx*tsi^3.0); % partial
derivative of coef
       coef1 = coef / psisq^3.5;    %
```

```
        dcoef1dx=(dcoefdx/psisq^3.5)-3.5*(coef*dpsisqdx/psisq^4.5); % partial
derivative of coef1
        cc2   = coef1 * no * (ao * (1.0 + 1.5 * etasq + eeta *...
            (4.0 + etasq)) + 0.375 * j2 * tsi / psisq * con41 *...
            (8.0 + 3.0 * etasq * (8.0 + etasq)));    %
        dcc2dx=(dcoef1dx*no+coef1*dnodx)*(ao*(1.0+1.5*etasq+eeta*...
            (4.0+etasq))+0.375*j2*tsi/psisq*con41*...
            (8.0+3.0*etasq*(8.0+etasq)))...
            +coef1*no*(daodx*(1.0+1.5*etasq+eeta*(4.0+etasq))+...
            ao*(1.5*detasqdx+deetadx*(4.0+etasq)+eeta*(detasqdx))+...
            0.375*j2*dtsidx/psisq*con41*(8.0+3.0*etasq*(8.0+etasq))...
            -0.375*j2*tsi*dpsisqdx/(psisq^2)*con41*...
            (8.0+3.0*etasq*(8.0+etasq))+...
            0.375*j2*tsi/psisq*dcon41dx*(8.0+3.0*etasq*(8.0+etasq))+...
            0.375*j2*tsi/psisq*con41*(3.0*detasqdx*(8.0+etasq)+...
            3.0*etasq*(detasqdx))); % partial derivative of cc2

        cc1   = bstar * cc2; %*
        dcc1dx=dbstardx*cc2+bstar*dcc2dx; % partial derivative of cc1
        cc3   = 0.0; %
        dcc3dx=[0,0,0,0,0,0,0]; % partial derivative of cc3
        if (ecco > 1.0e-4)
            cc3 = -2.0 * coef * tsi * j3oj2 * no * sinio / ecco; %
            dcc3dx=-2.0*dcoefdx*tsi*j3oj2*no*sinio/ecco...
                -2.0*coef*dtsidx*j3oj2*no*sinio/ecco...
                -2.0*coef*tsi*j3oj2*dnodx*sinio/ecco...
                -2.0*coef*tsi*j3oj2*no*dsiniodx/ecco...
                +2.0*coef*tsi*j3oj2*no*sinio*deccodx/(ecco^2); % partial
derivative of cc3
        end
        x1mth2 = 1.0 - cosio2;    %
        dx1mth2dx=-dcosio2dx; % partial derivative of x1mth2
        cc4    = 2.0* no * coef1 * ao * omeosq *...
            (eta * (2.0 + 0.5 * etasq) + ecco *...
            (0.5 + 2.0 * etasq) - j2 * tsi / (ao * psisq) *...
            (-3.0 * con41 * (1.0 - 2.0 * eeta + etasq *...
            (1.5 - 0.5 * eeta)) + 0.75 * x1mth2 *...
            (2.0 * etasq - eeta * (1.0 + etasq)) * cos(2.0 * argpo)));
        dcc4dx=2.0*(dnodx*coef1*ao*omeosq+no*dcoef1dx*ao*omeosq+...
            no*coef1*daodx*omeosq+no*coef1*ao*domeosqdx)*...
            (eta * (2.0 + 0.5 * etasq) + ecco *...
            (0.5 + 2.0 * etasq) - j2 * tsi / (ao * psisq) *...
            (-3.0 * con41 * (1.0 - 2.0 * eeta + etasq *...
            (1.5 - 0.5 * eeta)) + 0.75 * x1mth2 *...
            (2.0 * etasq - eeta * (1.0 + etasq)) *...
            cos(2.0 * argpo)))+...
            2.0*(no * coef1 * ao * omeosq )*...
            (detadx*(2.0+0.5*etasq)+eta*(0.5*detasqdx)+...
            deccodx*(0.5+2.0*etasq)+ecco*(2.0*detasqdx)...
            -j2*(dtsidx/(ao*psisq)-tsi*(daodx*psisq+ao*dpsisqdx)/...
            ((ao*psisq)^2))*(-3.0*con41*(1.0-2.0*eeta+etasq*...
            (1.5-0.5*eeta))+0.75*x1mth2*...
            (2.0*etasq-eeta*(1.0+etasq))*cos(2.0*argpo))...
            -j2*(tsi/(ao*psisq))*(-3.0*dcon41dx*(1.0-2.0*eeta+etasq*...
            (1.5-0.5*eeta))-3.0*con41*(-2.0*deetadx+detasqdx*...
            (1.5-0.5*eeta)+etasq*(-0.5*deetadx))+...
            0.75*dx1mth2dx*(2.0*etasq-eeta*(1.0+etasq))*...
```

```
           cos(2.0*argpo)+0.75*x1mth2*(2.0*detasqdx-deetadx*...
           (1.0+etasq)-eeta*(detasqdx))*cos(2.0*argpo)-0.75*x1mth2*...
           (2.0*etasq-eeta*(1.0+etasq))*sin(2.0*argpo)*2.0*dargpodx)); %
partial derivative of cc4

       cc5 = 2.0 * coef1 * ao * omeosq * (1.0 + 2.75 *...
           (etasq + eeta) + eeta * etasq);  %*
       dcc5dx=2.0*dcoef1dx*ao*omeosq*(1.0+2.75*(etasq+eeta)+...
           eeta*etasq)+2.0*coef1*daodx*omeosq*(1.0+2.75*...
           (etasq+eeta)+eeta*etasq)+2.0*coef1*ao*domeosqdx*...
           (1.0+2.75*(etasq+eeta)+eeta*etasq)+2.0*coef1*ao*omeosq*...
           (2.75*(detasqdx+deetadx)+deetadx*etasq+eeta*detasqdx); % partial
derivative of cc5
       cosio4 = cosio2 * cosio2;    %
       dcosio4dx=2*cosio2*dcosio2dx; % partial derivative of cosio4
       temp1  = 1.5 * j2 * pinvsq * no; %
       dtemp1dx=1.5*j2*(dpinvsqdx*no+pinvsq*dnodx); % partial derivative of
temp1
       temp2  = 0.5 * temp1 * j2 * pinvsq;  %
       dtemp2dx=0.5*dtemp1dx*j2*pinvsq+0.5*temp1*j2*dpinvsqdx; % partial
derivative of temp2
       temp3  = -0.46875 * j4 * pinvsq * pinvsq * no;   %
       dtemp3dx=-0.46875*j4*...
           (2*pinvsq*dpinvsqdx*no+pinvsq*pinvsq*dnodx); % partial derivative
of temp3
       mdot     = no + 0.5 * temp1 * rteosq * con41 +...
           0.0625 * temp2 * rteosq * (13.0 - 78.0 * cosio2 +...
           137.0 * cosio4);
       dmdotdx=dnodx+0.5*(dtemp1dx*rteosq*con41+...
           temp1*drteosqdx*con41+temp1*rteosq*dcon41dx)+...
           0.0625*dtemp2dx*rteosq*(13.0-78.0*cosio2+137.0*cosio4)+...
           0.0625*temp2*drteosqdx*(13.0-78.0*cosio2+137.0*cosio4)+...
           0.0625*temp2*rteosq*(-78.0*dcosio2dx+137.0*dcosio4dx); % partial
derivative of mdot
       argpdot  = -0.5 * temp1 * con42 + 0.0625 * temp2 *...
           (7.0 - 114.0 * cosio2 + 395.0 * cosio4) +...
           temp3 * (3.0 - 36.0 * cosio2 + 49.0 * cosio4);   %*
       dargpdotdx=-0.5*dtemp1dx*con42-0.5*temp1*dcon42dx+...
           0.0625*dtemp2dx*(7.0-114.0*cosio2+395.0*cosio4)+...
           0.0625*temp2*(-114.0*dcosio2dx+395.0*dcosio4dx)+...
           dtemp3dx*(3.0-36.0*cosio2+49.0*cosio4)+...
           temp3*(-36.0*dcosio2dx+49.0*dcosio4dx); % partial derivative of
argpdot
       xhdot1            = -temp1 * cosio;  %
       dxhdot1dx=-dtemp1dx*cosio-temp1*dcosiodx; % partial derivative of
xhdot1
       nodedot = xhdot1 + (0.5 * temp2 * (4.0 - 19.0 * cosio2) +...
           2.0 * temp3 * (3.0 - 7.0 * cosio2)) * cosio; %*
       dnodedotdx=dxhdot1dx+(0.5*dtemp2dx*(4.0-19.0*cosio2)+...
           0.5*temp2*(-19.0*dcosio2dx)+2.0*dtemp3dx*(3.0-...
           7.0*cosio2)+2.0*temp3*(-7.0*dcosio2dx))*cosio+...
           (0.5*temp2*(4.0-19.0*cosio2)+...
           2.0*temp3*(3.0-7.0*cosio2))*dcosiodx; % partial derivative of
nodedot
       omgcof   = bstar * cc3 * cos(argpo); %*
       domgcofdx=dbstardx*cc3*cos(argpo)+bstar*dcc3dx*cos(argpo)-...
           bstar*cc3*sin(argpo)*dargpodx; % partial derivative of omgcof
       xmcof    = 0.0;  %*
       dxmcofdx=[0,0,0,0,0,0,0,0]; % partial derivative of xmcof
       if (ecco > 1.0e-4)
           xmcof = -x2o3 * coef * bstar / eeta; %*
           dxmcofdx=-x2o3*((dcoefdx*bstar/eeta)+...
               (coef*dbstardx/eeta)-(coef*bstar*deetadx/(eeta^2))); % partial
derivative of xmcof
       end
       nodecf = 3.5 * omeosq * xhdot1 * cc1;    %*
       dnodecfdx=3.5*(domeosqdx*xhdot1*cc1+omeosq*dxhdot1dx*cc1+...
           omeosq*xhdot1*dcc1dx); % partial derivative of nodecf
       t2cof   = 1.5 * cc1; %*
       dt2cofdx=1.5*dcc1dx; % partial derivative of t2cof

%        % // sgp4fix for divide by zero with xinco = 180 deg
%        if (abs(cosio+1.0) > 1.5e-12)
%           xlcof   = -0.25 * j3oj2 * sinio *...
%               (3.0 + 5.0 * cosio) / (1.0 + cosio);
%        else
%           xlcof   = -0.25 * j3oj2 * sinio *...
%               (3.0 + 5.0 * cosio) / temp4;
%        end
%        aycof   = -0.5 * j3oj2 * sinio;
       delmo   = (1.0 + eta * cos(mo))^3;   %*
       ddelmodx=3*((1.0 + eta * cos(mo))^2)*(detadx*cos(mo)-...
           eta*sin(mo)*dmodx); % partial derivative of delmo
       sinmao  = sin(mo);   %*
       dsinmaodx=cos(mo)*dmodx; % partial derivative of sinmao

       % /* --------------- deep space initialization ------------- */
       if ((2*pi / no) >= 225.0)
           method = 'd';
           isimp  = 1;
       end

       %initialize variables and partial derivatives
       d2=0.0;
       dd2dx=[0,0,0,0,0,0,0,0]; % partial derivative of d2
       d3=0.0;
       dd3dx=[0,0,0,0,0,0,0,0]; % partial derivative of d3
       d4=0.0;
       dd4dx=[0,0,0,0,0,0,0,0]; % partial derivative of d4
       t3cof=0.0;
       dt3cofdx=[0,0,0,0,0,0,0,0]; % partial derivative of t3cof
       t4cof=0.0;
       dt4cofdx=[0,0,0,0,0,0,0,0]; % partial derivative of t4cof
       t5cof=0.0;
       dt5cofdx=[0,0,0,0,0,0,0,0]; % partial derivative of t5cof

       % /* ----------- set variables and partial derivatives if not deep
space ----------- */
       if (isimp ~= 1)
           cc1sq          = cc1 * cc1;  %
           dcc1sqdx=2*cc1*dcc1dx; % partial derivative of cc1sq
           d2    = 4.0 * ao * tsi * cc1sq;  %*
           dd2dx=4.0*(daodx*tsi*cc1sq+ao*dtsidx*cc1sq+ao*tsi*dcc1sqdx); %
partial derivative of d2
           temp           = d2 * tsi * cc1 / 3.0;   %
           dtempdx=(dd2dx*tsi*cc1+d2*dtsidx*cc1+d2*tsi*dcc1dx)/ 3.0; %
partial derivative of temp
           d3    = (17.0 * ao + sfour) * temp;  %*
           dd3dx=(17.0*daodx+dsfourdx)*temp+(17.0*ao+sfour)*dtempdx; %
partial derivative of d3
           d4    = 0.5 * temp * ao * tsi *...
               (221.0 * ao + 31.0 * sfour) * cc1;   %*
           dd4dx=0.5*dtempdx*ao*tsi*(221.0*ao+31.0*sfour)*cc1+...
               0.5*temp*daodx*tsi*(221.0*ao+31.0*sfour)*cc1+...
               0.5*temp*ao*dtsidx*(221.0*ao+31.0*sfour)*cc1+...
               0.5*temp*ao*tsi*(221.0*daodx+31.0*dsfourdx)*cc1+...
               0.5*temp*ao*tsi*(221.0*ao+31.0*sfour)*dcc1dx; % partial
derivative of d4
           t3cof = d2 + 2.0 * cc1sq;    %*
           dt3cofdx=dd2dx+2.0*dcc1sqdx; % partial derivative of t3cof
           t4cof = 0.25 * (3.0 * d3 + cc1 *...
               (12.0 * d2 + 10.0 * cc1sq));     %*
           dt4cofdx=0.25*(3.0*dd3dx+dcc1dx*(12.0*d2+10.0*cc1sq)+ ...
               cc1*(12.0*dd2dx+10.0*dcc1sqdx)); % partial derivative of t4cof
           t5cof = 0.2 * (3.0 * d4 +...
               12.0 * cc1 * d3 +...
               6.0 * d2 * d2 +...
               15.0 * cc1sq * (2.0 * d2 + cc1sq));   %*
           dt5cofdx=0.2 * (3.0*dd4dx+12.0*dcc1dx*d3+12.0*cc1*dd3dx+...
               6.0*2*d2*dd2dx+15.0*dcc1sqdx*(2.0*d2+cc1sq)+ ...
               15.0*cc1sq*(2.0*dd2dx+dcc1sqdx)); % partial derivative of
t5cof
       end
   end % // if omeosq = 0 ...


%% ------- update for secular gravity and atmospheric drag -----
   xmdf    = mo + mdot * t;
   dxmdfdx=dmodx+dmdotdx*t; % partial derivative of xmdf
   argpdf  = argpo + argpdot * t;
   dargpdfdx=dargpodx+dargpdotdx*t; % partial derivative ofargpdf
   nodedf  = nodeo + nodedot * t;
   dnodedfdx=dnodeodx+dnodedotdx*t; % partial derivative ofnodedf
   argpm   = argpdf;
   dargpmdx=dargpdfdx; % partial derivative of argpm
   mm      = xmdf;
   dmmdx=dxmdfdx; % partial derivative of mm
   t2      = t * t;
   nodem   = nodedf + nodecf * t2;
   dnodemdx=dnodedfdx+dnodecfdx*t2; % partial derivative of nodem
   tempa   = 1.0 - cc1 * t;
   dtempadx=-dcc1dx*t; % partial derivative of tempa
   tempe   = bstar * cc4 * t;
   dtempedx=dbstardx*cc4*t+bstar*dcc4dx*t; % partial derivative of tempe
   templ   = t2cof * t2;
   dtempldx=dt2cofdx*t2; % partial derivative of templ

   if (isimp ~= 1) %not deepspace
```

```
       delomg = omgcof * t;
       ddelomgdx=domgcofdx*t; % partial derivative of delomg
       delm   = xmcof *...
           ((1.0 + eta * cos(xmdf))^3 -...
           delmo);
       ddelmdx=dxmcofdx*(((1.0+eta*cos(xmdf))^3)-delmo)+ ...
           xmcof*(3*((1.0 + eta * cos(xmdf))^2)* ...
           (detadx*cos(xmdf)-eta*sin(xmdf)*dxmdfdx)-ddelmodx); % partial
derivative of delm
       temp   = delomg + delm;
       dtempdx=ddelomgdx+ddelmdx; % partial derivative of temp
       mm     = xmdf + temp;
       dmmdx=dxmdfdx+dtempdx; % partial derivative of mm
       argpm  = argpdf - temp;
       dargpmdx=dargpdfdx-dtempdx; % partial derivative of argpm
       t3     = t2 * t;
       t4     = t3 * t;
       tempa  = tempa - d2 * t2 - d3 * t3 -...
           d4 * t4;
       dtempadx=dtempadx-dd2dx*t2-dd3dx*t3-dd4dx*t4; % partial derivative of
tempa
       tempe  = tempe + bstar * cc5 * (sin(mm) -...
           sinmao);
       dtempedx=dtempedx+dbstardx*cc5*(sin(mm)-sinmao)+ ...
           bstar*dcc5dx*(sin(mm)-sinmao)+ ...
           bstar*cc5*(cos(mm)*dmmdx-dsinmaodx); % partial derivative of tempe
       templ  = templ + t3cof * t3 + t4 * (t4cof +...
           t * t5cof);
       dtempldx=dtempldx+dt3cofdx*t3+t4*(dt4cofdx+t*dt5cofdx); % partial
derivative of templ
   end

   nm     = no;
   dnmdx=dnodx; % partial derivative of nm
   em     = ecco;
   demdx=deccodx; % partial derivative of em

   if (method == 'd')
       error('error satellite requires SDP4');
   end % // if method = d

   if (nm <= 0.0)
%       fprintf(1,'# error nm %f\n', nm);
       saterror = 2;
   end

   am = (xke / nm)^x2o3 * tempa * tempa; % new value for semi-major axis
   damdx=2*sqrt(am)*(((-1/3)*(xke/(nm^4))^(1/3))* ...
       tempa*dnmdx+dtempadx*(xke/nm)^(1/3)); % partial derivative of am
   em = em - tempe; % new value for eccentricity
   demdx=demdx-dtempedx; % partial derivative of em

   % // fix tolerance for error recognition
   if ((em >= 1.0) || (em < -0.001) || (am < 0.95))
%       fprintf(1,'# error em %f\n', em);
       saterror = 1;
   end
%   sgp4fix change test condition for eccentricity
   if (em < 1.0e-6)
       em  = 1.0e-6;
   end
   mm      = mm + no * templ;
   dmmdx=dmmdx+(no*dtempldx+dnodx*templ); % partial derivative of mm
   xlm     = mm + argpm + nodem;
   dxlmdx=dmmdx+dargpmdx+dnodemdx; % partial derivative of xlm

   nodem  = rem(nodem, twopi); % normalize RAAN
   argpm  = rem(argpm, twopi); % normalize argument of perigee
   xlm    = rem(xlm, twopi);
   mm     = rem(xlm - argpm - nodem, twopi); % new value for mean anomaly
   dmmdx=dxlmdx-dargpmdx-dnodemdx; % partial derivative of mm

%% output

y = [am;em;inclo;argpm;nodem;mm;bstar]; % new values for mean elements
dy= [damdx;demdx;dinclodx;dargpmdx;dnodemdx;dmmdx;dbstardx]; % partial
derivatives of the secular variations of the mean elements

end
```

Short and Long Period Variations

```
function [ r,v,dr,dv,saterror ] = sgp4slpd( x,method )
% performs short and long period perturbation update from the Brouwer mean
orbital elements using the near earth portion of the sgp4 propagator
%
% --------------------------------------------------------------------
% Author:
%   Thomas Wright
% original matlab code:
%   Jeff Beck
%   beckja@alumni.lehigh.edu
%    current :
%                7 may 08  david vallado
%                            update small eccentricity check
%    changes :
%               16 nov 07  david vallado
%                            misc fixes for better compliance
%   1.0 (aug 7, 2006) - update for paper dav
% original Vallado C++ version:
%   author        : david vallado          719-573-2600   28 jun 2005
%   references    :
%     hoots, roehrich, norad spacetrack report #3 1980
%     hoots, norad spacetrack report #6 1986
%     hoots, schumacher and glover 2004
%     vallado, crawford, hujsak, kelso  2006
% --------------------------------------------------------------------
% INPUTS:
%   x   a 1 by 7 array of orbital elements in the following order
%       (1) ao      Earth radii     mean semi-major axis
%       (2) ecco    unitless        mean eccentricity
%       (3) inclo   radians         mean inclination
%       (4) argpo   radians         mean argument of perigee
%       (5) nodeo   radians         mean RAAN
%       (6) mo      radians         mean mean anomaly
%       (7) bstar   1/Earth radius  drag coefficient
%   method  'n'/'d' 'n'ear earth or 'd'eep space propagator
%
% OUTPUTS:
%   r   km      instantaneous position in TEME coordinates
%   v   km/s    instantaneous velocity in TEME coordinates
%   dr  (3x7)   Jacobian of SGP4 TEME position with respect to the orbital
elements and B*
%   dv  (3x7)   Jacobian of SGP4 TEME velocity with respect to the orbital
elements and B*
%   saterror    error code:
%                   0 - no error
%                   1 - mean elements, ecc >= 1.0 or ecc < -0.001 or
%                      a < 0.95
%                   2 - mean motion less than 0.0
%                   3 - pert elements, ecc < 0.0  or  ecc > 1.0
%                   4 - semi-latus rectum < 0.0
%                   5 - epoch elements are sub-orbital
%                   6 - satellite has decayed
%

%% /* ------------------ set mathematical constants --------------- */
   twopi = 2.0 * pi;
   x2o3  = 2.0 / 3.0;
   % sgp4fix divisor for divide by zero check on inclination
   % the old check used 1.0 + cos(pi-1.0e-9), but then compared it to
   % 1.5 e-12, so the threshold was changed to 1.5e-12 for consistancy
   temp4    =   1.5e-12;    %

   %  // sgp4fix identify constants and allow alternate values
   global tumin mu radiusearthkm xke j2 j3 j4 j3oj2
   vkmpersec     = radiusearthkm * xke/60.0;    %

%% /* --------------------- clear sgp4 error flag ----------------- */
   saterror = 0;
   mrt = 0.0;

method = method;
am     = x(1);
damdx=[1,0,0,0,0,0,0]; % partial derivative of am
nm     = xke/(am^1.5);
dnmdx=-1.5*xke*damdx/(am^2.5); % partial derivative of nm
em     = x(2);
demdx=[0,1,0,0,0,0,0]; % partial derivative of em
inclm  = x(3);
dinclmdx=[0,0,1,0,0,0,0]; % partial derivative of inclm
argpm  = x(4);
dargpmdx=[0,0,0,1,0,0,0]; % partial derivative of argpm
nodem  = x(5);
dnodemdx=[0,0,0,0,1,0,0]; % partial derivative of nodem
mm     = x(6);
dmmdx=[0,0,0,0,0,1,0]; % partial derivative of mm
```

```
%% initializations and their partial derivatives
cosio  = cos(inclm);
dcosiodx=-sinio*dinclmdx; % partial derivative of cosio
cosio2 = cosio * cosio;
dcosio2dx=2*cosio*dcosiodx; % partial derivative of cosio2
sinio = sin(inclm);
dsiniodx=cosio*dinclmdx; % partial derivative ofsinio

con42 = 1.0 - 5.0 * cosio2;
dcon42dx=-5.0*dcosio2dx; % partial derivative of con42
con41 = -con42-cosio2-cosio2;
dcon41dx=-dcon42dx-dcosio2dx-dcosio2dx; % partial derivative of con41

eccsq  = em * em;
omeosq = 1.0 - eccsq;
   if ((omeosq >= 0.0 ) | ( nm >= 0.0))
       x1mth2 = 1.0 - cosio2;   %
       dx1mth2dx=-dcosio2dx; % partial derivative of x1mth2

       % // sgp4fix for divide by zero with xinco = 180 deg
       if (abs(cosio+1.0) > 1.5e-12)
          xlcof   = -0.25 * j3oj2 * sinio *...
               (3.0 + 5.0 * cosio) / (1.0 + cosio);
       else
          xlcof   = -0.25 * j3oj2 * sinio *...
               (3.0 + 5.0 * cosio) / temp4;
       end
       aycof   = -0.5 * j3oj2 * sinio;
       daycofdx=-0.5*j3oj2*dsiniodx; % partial derivative of aycof

       x7thm1  = 7.0 * cosio2 - 1.0;
       dx7thm1dx=7.0*dcosio2dx; % partial derivative of x7thm1
   end

%% /* ----------------- compute extra mean quantities and their partial
derivatives ------------- */
   sinim = sin(inclm);  %
   dsinimdx=cosim*dinclmdx; % partial derivative of sinim
   cosim = cos(inclm);  %
   dcosimdx=-sinim*dinclmdx; % partial derivative of cosim

%% /* -------------------- add lunar-solar periodics -------------- */
   ep     = em;     %
   depdx=demdx; % partial derivative of ep
   xincp  = inclm;  %
   dxincpdx=dinclmdx; % partial derivative of xincp
   argpp  = argpm;  %
   dargppdx=dargpmdx; % partial derivative of argpp
   nodep  = nodem;  %
   dnodepdx=dnodemdx; % partial derivative of nodep
   mp     = mm;     %
   dmpdx=dmmdx; % partial derivative of mp
   sinip  = sinim;  %
   dsinipdx=dsinimdx; % partial derivative of sinip
   cosip  = cosim;  %
   dcosipdx=dcosimdx; % partial derivative of cosip
   if (method == 'd')   % deepspace
       error('error satellite requires SDP4');
   end % // if method = d

   % /* -------------------- long period periodics ------------------ */
   if (method == 'd')   % deepspace
       error('error satellite requires SDP4');
   end
   axnl = ep * cos(argpp);  %
   daxnldx=depdx*cos(argpp)-ep*sin(argpp)*dargppdx; % partial derivative of
axnl
   temp = 1.0 / (am * (1.0 - ep * ep)); %
   dtempdx=-(damdx*(1.0-ep*ep)+am*(-2*ep*depdx))/((am*(1.0-ep*ep))^2); %
partial derivative of temp
   aynl = ep* sin(argpp) + temp * aycof;    %
   daynldx=depdx*sin(argpp)+ep*cos(argpp)*dargppdx+...
       dtempdx*aycof+temp*daycofdx; % partial derivative of aynl
   xl   = mp + argpp + nodep + temp * xlcof * axnl; %

%% /* --------------------- solve kepler's equation --------------- */
   u    = rem(xl - nodep, twopi);   %
   eo1  = u;    %
   tem5 = 9999.9;   %
   ktr = 1; %
   % //   sgp4fix for kepler iteration
   % //   the following iteration needs better limits on corrections
   while (( abs(tem5) >= 1.0e-12) && (ktr <= 10) )
       sineo1 = sin(eo1);   %
       coseo1 = cos(eo1);   %
       tem5   = 1.0 - coseo1 * axnl - sineo1 * aynl;    %
       tem5   = (u - aynl * coseo1 + axnl * sineo1 - eo1) / tem5;   %
       if(abs(tem5) >= 0.95)
           if tem5 > 0.0
               tem5 = 0.95; %
           else
               tem5 = -0.95;    %
           end
       end
       eo1    = eo1 + tem5; %
       ktr = ktr + 1;   %
   end
   % partial derivatives of kepler’s equation
   deo1dx=(depdx*(sineo1*cos(argpp)-coseo1*sin(argpp))+dmpdx)/ ...
       (1-ep*(coseo1*cos(argpp)+sineo1*sin(argpp)))+dargppdx;
   dsineo1dx=coseo1*deo1dx;
   dcoseo1dx=-sineo1*deo1dx;

%% /* ------------- short period preliminary quantities ----------- */
   ecose = axnl*coseo1 + aynl*sineo1;   %
   decosedx=daxnldx*coseo1+axnl*dcoseo1dx+daynldx*sineo1+...
       aynl*dsineo1dx; % partial derivative of ecose
   esine = axnl*sineo1 - aynl*coseo1;   %
   desinedx=daxnldx*sineo1+axnl*dsineo1dx-daynldx*coseo1-...
       aynl*dcoseo1dx; % partial derivative of esine
   el2   = axnl*axnl + aynl*aynl;   %
   del2dx=2*axnl*daxnldx+2*aynl*daynldx; % partial derivative of el2
   pl    = am*(1.0-el2);    %
   dpldx=damdx*(1.0-el2)-am*del2dx; % partial derivative of pl
   if (pl < 0.0)
       saterror = 4;
       r = [0;0;0];
       v = [0;0;0];
   else
       rl     = am * (1.0 - ecose); %
       drldx=damdx*(1.0-ecose)-am*decosedx; % partial derivative of rl
       rdotl  = sqrt(am) * esine/rl;    %
       drdotldx=0.5*damdx*esine/(sqrt(am)*rl)+sqrt(am)*desinedx/rl-...
           sqrt(am)*esine*drldx/(rl^2); % partial derivative of rdotl
       rvdotl = sqrt(pl) / rl;  %
       drvdotldx=0.5*dpldx/(sqrt(pl)*rl)-sqrt(pl)*drldx/(rl^2); % partial
derivative of rvdotl
       betal  = sqrt(1.0 - el2);    %
       dbetaldx=-0.5*del2dx/sqrt(1.0 - el2); % partial derivative of betal
       temp   = esine / (1.0 + betal);  %
       dtempdx=desinedx/(1.0+betal)-esine*dbetaldx/((1.0 + betal)^2); %
partial derivative of temp
       sinu   = am / rl * (sineo1 - aynl - axnl * temp);    %
       dsinudx=damdx/rl*(sineo1-aynl-axnl*temp)- ...
           am*drldx/(rl^2)*(sineo1-aynl-axnl*temp)+ ...
           am/rl*(dsineo1dx-daynldx+daxnldx*temp+axnl*dtempdx); % partial
derivative of sinu
       cosu   = am / rl * (coseo1 - axnl + aynl * temp);    %
       dcosudx=damdx/rl*(coseo1-axnl+aynl*temp)- ...
           am*drldx/(rl^2)*(coseo1-axnl+aynl*temp)+ ...
           am/rl*(dcoseo1dx-daxnldx+daynldx*temp+aynl*dtempdx); % partial
derivative of cosu
       su     = atan2(sinu, cosu);  %
       dsudx=(dsinudx*cosu-sinu*dcosudx)/(cosu*cosu+sinu*sinu); % partial
derivative of su
       sin2u  = (cosu + cosu) * sinu;   %
       dsin2udx=(dcosudx+dcosudx)*sinu+(cosu+cosu)*dsinudx; % partial
derivative of sin2u
       cos2u  = 1.0 - 2.0 * sinu * sinu;    %
       dcos2udx=-4.0*sinu*dsinudx; % partial derivative of cos2u
       temp   = 1.0 / pl;   %
       dtempdx=-dpldx/(pl^2); % partial derivative of temp
       temp1  = 0.5 * j2 * temp;    %
       dtemp1dx=0.5*j2*dtempdx; % partial derivative of temp1
       temp2  = temp1 * temp;   %
       dtemp2dx=dtemp1dx*temp+temp1*dtempdx; % partial derivative of temp2

%% /* -------------- update for short period periodics ------------ */
       if (method == 'd')    % deepspace
           error('error satellite requires SDP4');
       end
       mrt   = rl * (1.0 - 1.5 * temp2 * betal * con41) +...
           0.5 * temp1 * x1mth2 * cos2u;     %
       dmrtdx=drldx*(1.0-1.5*temp2*betal*con41)+ ...
           rl*(-1.5*dtemp2dx*betal*con41-1.5*temp2*dbetaldx*con41-...
           1.5*temp2*betal*dcon41dx)+...
           0.5*dtemp1dx*x1mth2*cos2u+0.5*temp1*dx1mth2dx*cos2u+...
           0.5*temp1*x1mth2*dcos2udx; % partial derivative of mrt
       su    = su - 0.25 * temp2 * x7thm1 * sin2u;   %
       dsudx=dsudx-0.25*dtemp2dx*x7thm1*sin2u-...
           0.25*temp2*dx7thm1dx*sin2u-0.25*temp2*x7thm1*dsin2udx; % partial
derivative of su
       xnode = nodep + 1.5 * temp2 * cosip * sin2u;     %
       dxnodedx=dnodepdx+1.5*dtemp2dx*cosip*sin2u+...
           1.5*temp2*dcosipdx*sin2u+1.5*temp2*cosip*dsin2udx; % partial
derivative of xnode
       xinc  = xincp + 1.5 * temp2 * cosip * sinip * cos2u;     %
       dxincdx=dxincpdx+1.5*dtemp2dx*cosip*sinip*cos2u+...
           1.5*temp2*dcosipdx*sinip*cos2u+...
           1.5*temp2*cosip*dsinipdx*cos2u+...
           1.5*temp2*cosip*sinip*dcos2udx; % partial derivative of xinc
       mvt   = rdotl - nm * temp1 * x1mth2 * sin2u / xke;    %
       dmvtdx=drdotldx-dnmdx*temp1*x1mth2*sin2u/xke-...
           nm*dtemp1dx*x1mth2*sin2u/xke-...
           nm*temp1*dx1mth2dx*sin2u/xke-nm*temp1*x1mth2*dsin2udx/xke; %
partial derivative of mvt
       rvdot = rvdotl + nm * temp1 * (x1mth2 * cos2u +...
           1.5 * con41) / xke;   %
       drvdotdx=drvdotldx+dnmdx*temp1*(x1mth2*cos2u+1.5*con41)/xke+...
           nm*dtemp1dx*(x1mth2*cos2u+1.5*con41)/xke+...
           nm*temp1*(dx1mth2dx*cos2u+x1mth2*dcos2udx+1.5*dcon41dx)/xke; %
partial derivative of rvdot

%% /* --------------------- orientation vectors and their partial derivatives
-------------------- */
       sinsu =  sin(su);    %
       dsinsudx=cossu*dsudx; % partial derivative of sinsu
       cossu =  cos(su);    %
       dcossudx=-sinsu*dsudx; % partial derivative of cossu
       snod  =  sin(xnode); %
       dsnoddx=cnod*dxnodedx; % partial derivative of snod
       cnod  =  cos(xnode); %
       dcnoddx=-snod*dxnodedx; % partial derivative of cnod
       sini  =  sin(xinc);  %
       dsinidx=cosi*dxincdx; % partial derivative of sini
       cosi  =  cos(xinc);  %
       dcosidx=-sini*dxincdx; % partial derivative of cosi
       xmx   = -snod * cosi;    %
       dxmxdx=-dsnoddx*cosi-snod*dcosidx; % partial derivative of xmx
       xmy   =  cnod * cosi;    %
       dxmydx=dcnoddx*cosi+cnod*dcosidx; % partial derivative of xmy
       ux    =  xmx * sinsu + cnod * cossu; %
       duxdx=dxmxdx*sinsu+xmx*dsinsudx+dcnoddx*cossu+cnod*dcossudx; % partial
derivative of ux
       uy    =  xmy * sinsu + snod * cossu; %
       duydx=dxmydx*sinsu+xmy*dsinsudx+dsnoddx*cossu+snod*dcossudx; % partial
derivative of uy
       uz    =  sini * sinsu;   %
       duzdx=dsinidx*sinsu+sini*dsinsudx; % partial derivative of uz
       vx    =  xmx * cossu - cnod * sinsu; %
       dvxdx=dxmxdx*cossu+xmx*dcossudx-dcnoddx*sinsu-cnod*dsinsudx; % partial
derivative of vx
       vy    =  xmy * cossu - snod * sinsu; %
       dvydx=dxmydx*cossu+xmy*dcossudx-dsnoddx*sinsu-snod*dsinsudx; % partial
derivative of vy
       vz    =  sini * cossu;   %
       dvzdx=dsinidx*cossu+sini*dcossudx; % partial derivative ofvz
%% /* --------- position and velocity (in km and km/sec) and their partial
derivatives ---------- */
       r = [0;0;0];
       v = [0;0;0];
       r(1) = (mrt * ux)* radiusearthkm;
       dr1dx=(dmrtdx*ux+mrt*duxdx)* radiusearthkm; % partial derivative of
r(1)
       r(2) = (mrt * uy)* radiusearthkm;
       dr2dx=(dmrtdx*uy+mrt*duydx)* radiusearthkm; % partial derivative of
r(2)
       r(3) = (mrt * uz)* radiusearthkm;
       dr3dx=(dmrtdx*uz+mrt*duzdx)* radiusearthkm; % partial derivative of
r(3)
       v(1) = (mvt * ux + rvdot * vx) * vkmpersec;
       dv1dx=(dmvtdx*ux+mvt*duxdx+drvdotdx*vx+rvdot*dvxdx)*vkmpersec; %
partial derivative of v(1)
       v(2) = (mvt * uy + rvdot * vy) * vkmpersec;
       dv2dx=(dmvtdx*uy+mvt*duydx+drvdotdx*vy+rvdot*dvydx)*vkmpersec; %
partial derivative of v(2)
       v(3) = (mvt * uz + rvdot * vz) * vkmpersec;
       dv3dx=(dmvtdx*uz+mvt*duzdx+drvdotdx*vz+rvdot*dvzdx)*vkmpersec; %
partial derivative of v(3)

       dr=[dr1dx;dr2dx;dr3dx]; % partial derivative of the position vector
       dv=[dv1dx;dv2dx;dv3dx]; % partial derivative of the velocity vector
   end % // if pl > 0

       % // sgp4fix for decaying satellites
        if (mrt < 1.0)
            saterror = 6;
        end

end
```