

**Faculty of Applied Science
University of Toronto**

**MOST Microsatellite Project
State Estimation using Kalman Filter
Final Report**

May 19, 2000

Report by: Charles Lidstone

Stud. No.: 920545400

Abstract

An extended Kalman filter algorithm for attitude estimation based on Earth's magnetic field measurements is discussed. A fixed gain state observer is implemented. The observer algorithm was simulated with real satellite magnetometer data from UoSAT-OSCAR-22 to verify the algorithms validity.

Table of Contents

Abstract	i
Table of Contents.....	ii
List of Symbols.....	iv
1. Introduction.....	1
2. Attitude Kinematics & Dynamics	2
2.1 Coordinate Systems.....	2
2.1.1 <i>Inertial Coordinate System (Inertial CS)</i>	2
2.1.2 <i>Orbit Coordinate System (Orbit CS)</i>	2
2.1.3 <i>Control Coordinate System (Control CS)</i>	3
2.2 Attitude Representation (Quaternion)	3
2.2.1 <i>Introduction to Quaternions</i>	4
2.2.2 <i>Quaternions as Rotations</i>	5
2.3 Equations of Motion.....	6
2.4 Numerical Methods.....	7
3. State Estimation	8
3.1 State Model.....	8
3.2 Observers	8
3.2.1 <i>Identity Observer</i>	9
3.2.2 <i>Current Estimator</i>	9
3.3 Optimal Estimation	10
3.3.1 <i>Practical Kalman Filter Algorithm</i>	11
3.3.2 <i>Extended Kalman Filter</i>	11
4. Sensor Models.....	13
4.1 Rate Sensors.....	13
4.2 Magnetometers.....	13
4.3 Sun Sensors.....	13
4.4 Star Tracker.....	13
5. Magnetometers	14
5.1 IGRF Model.....	14

5.1.1	<i>Altitude-Latitude-Longitude Coordinate System (ALL CS)</i>	14
5.1.2	<i>North-East-Down Coordinate System (NED CS)</i>	14
5.2	State Estimation Based on Vector Measurements.....	15
5.2.1	<i>Kalman Filtering Based on Vector Measurements</i>	17
6.	UoSAT-OSCAR 22 (UO-22)	18
6.1	UO-22 Telemetry & Orbital Elements	18
6.1.1	<i>Magnetometer Readings</i>	18
6.1.2	<i>Orbit Propagation</i>	19
6.2	Attitude Kinematics & Dynamics of UO-22.....	20
6.2.1	<i>Control Coordinate System (Control CS)</i>	20
6.2.2	<i>Gravity Gradient Effect</i>	20
6.2.3	<i>Modified Equations of Motion</i>	21
6.2.4	<i>Moment of Inertia</i>	22
7.	Observer Experiments	24
7.1	Dynamics Simulation.....	24
7.2	Observer Simulation (Dipole Model).....	25
7.3	Observer Experiment (Telemetry and IGRF model).....	28
7.4	Discussion	29
7.5	Other Simulations	29
8.	Conclusions and Future Work	30
	Appendix A	31
	Appendix B	34
	Reference	38

List of Symbols

(Kinematics & Dynamics only)

- ${}^c\bar{\mathbf{q}}_I$ is the attitude quaternion of the Control CS with respect to the Inertial CS,
 ${}^c\bar{\mathbf{q}}_O$ is the attitude quaternion of the Control CS with respect to the Orbit CS,
 ${}^c\bar{\boldsymbol{\omega}}_{CI}$ is the four-vector representation of the angular velocity ${}^c\boldsymbol{\omega}_{CI}$,
 ${}^c\boldsymbol{\omega}_{CI}$ is the angular velocity of the Control CS relative to the Inertial CS
expressed in the Control CS
 ${}^c\boldsymbol{\omega}_{CO}$ is the angular velocity of the Control CS relative to the Orbit CS
expressed in the Control CS
- $\hat{\mathbf{I}}$ is the moment of inertia of the spacecraft, and
- ${}^c\mathbf{g}$ is the applied torque expressed in the Control CS.
 ${}^c\mathbf{g}_{gg}$ is the gravity gradient torque expressed in the Control CS.
 ${}^c\mathbf{k}_O$ is the z-axis unit vector for the Orbit CS expressed in the Control CS
 ω_o is the orbital rate
 $\mathbf{R}(\bar{\mathbf{q}})$ is the 3x3 rotation matrix formed with the quaternion $\bar{\mathbf{q}}$
 $[\bar{\mathbf{q}}]$ is the 4x4 matrix representation of quaternion left-multiplication

1. Introduction

Attitude determination will be a critical function on MOST. Like all satellites, MOST will begin its mission in an unpredictable tumble. Even so, the mission of the spacecraft, to observe micro oscillations in stars, requires it to maintain precise attitude for many weeks at a time. The attitude control system must be capable of driving MOST to its desired attitude from a wide variety of initial conditions and with great precision. The capabilities of the attitude control system are greatly affected by the quality of the measurements it uses. MOST will require many sensors working together to provide the attitude control system with enough information. One of those sensors is a set of magnetometers.

Magnetometers measure Earth's magnetic field vector in three component directions as seen from the spacecraft. The Earth's magnetic field vector as seen from the Earth for a particular point in space can be determined [13]. Comparing these two vectors yields information about the attitude of the spacecraft relative to the Earth. The information is incomplete, however, because it is not possible to determine the rotation of the spacecraft about the magnetic field vector direction. The direction of the Earth's magnetic field vector varies with position. By making use of many measurements as the spacecraft orbits, it is possible to completely determine spacecraft attitude. This report presents an algorithm for determining attitude from magnetometer measurements based on the extended Kalman filter (EKF). To verify the EKF algorithm presented, data from a real satellite was used, and the attitude estimate produced was compared against a real satellite.

In order to implement an EKF algorithm, the attitude kinematics & dynamics of the spacecraft must be known. Attitude kinematics & dynamics are presented in Section 2. State estimation theory and the EKF are developed in Section 3. Section 4 contains a brief discussion about how sensors on MOST could be used in an EKF algorithm. Section 5 presents the steps necessary to make use of magnetometer measurements in an EKF. Some modifications to the theory presented in Section 2 through Section 5 was necessary due to the properties of the satellite selected for comparison. Section 6 is a discussion of the satellite selected and the modifications made. Section 7 is a discussion of the results of simulations and experiments using the EKF algorithm developed. Finally, conclusions are drawn, and the direction of future work is suggested.

2. Attitude Kinematics & Dynamics

State estimation depends on knowledge of the satellite's attitude kinematics and dynamics. In order to discuss these principles, coordinate systems and coordinate representations must be defined. It is then possible to present the satellite equations of motion.

2.1 Coordinate Systems

The coordinate systems discussed here are related to Attitude Kinematics & Dynamics only. Several other coordinate systems will be used in relation to Earth's magnetic field vector determination. They will be discussed in Section 5.1.1 and 5.1.2. Coordinate systems associated with UO-22 are discussed in Section 6.2.1.

Attitude equations of motion do not depend on the location of coordinate frame origins. This simplifies the definition of the coordinate systems used in this section.

2.1.1 Inertial Coordinate System (Inertial CS)

The Inertial CS is a right orthogonal inertial coordinate system. The location of the origin is not significant; it will be taken at the center of mass of the satellite for simplicity. The z-axis is taken to be parallel to the rotation axis of the earth, positive north. The x-axis is parallel to the line which connects the center of the earth to the vernal equinox. The vernal equinox is the point where the equator meets the ecliptic on the first day of spring. The ecliptic is the great circle which lies in the earth's plane of rotation about the sun.

2.1.2 Orbit Coordinate System (Orbit CS)

The Orbit CS is a right orthogonal coordinate system. The origin is placed at the center of mass of the satellite. The z-axis points toward the center of the earth. The x-axis points in the orbit normal direction, which is the vector perpendicular to the plane of the satellites orbit about the earth, oriented in the right handed sense with respect to the orbit direction. The y-axis points in the direction of instantaneous velocity.

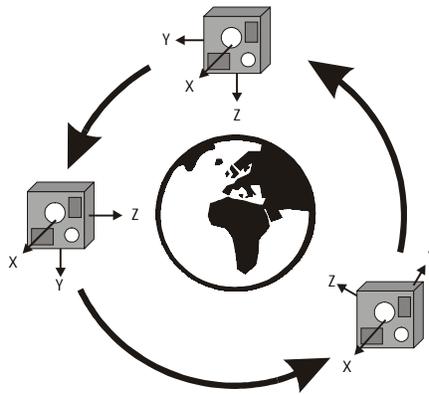


Figure 2.1: Orbit Coordinate System

2.1.3 Control Coordinate System (Control CS)

The Control CS is a right orthogonal coordinate system. The origin is placed at the center of mass of the satellite. The axes are aligned with the principal axis of the satellite's moment of inertia. The z-axis is chosen to point approximately in the direction of the science instrument.

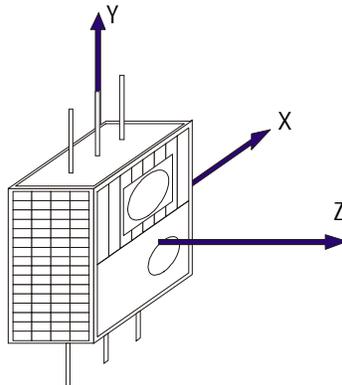


Figure 2.2: Control Coordinate System - MOST

2.2 Attitude Representation (Quaternion)

Traditionally, rotations are represented by Euler angles. Using the Euler angle method, the attitude of the spacecraft, which is the attitude of the Control CS, would be characterized by three angles, $\theta_x, \theta_y, \theta_z$, representing rotations about each of the x, y and z axes of the Inertial CS, respectively. Euler angles are easy to visualize for small rotations. For rotations of less than 20° , $\theta_x, \theta_y, \theta_z$ can be easily interpreted as yaw, pitch and roll,

respectively. For general orientations, not restricted to small angles, Euler angles have some disadvantages; most notably, the “gimbal lock” singularity. An example of “gimbal lock” occurs when performing the rotation $\theta_x, \theta_y = 90^\circ, \theta_z$. The rotation of 90° about the y-axis makes it impossible to distinguish between rotations about the x and z axes. Thus, attitude quaternion, an alternative representation, has been chosen for the attitude control system design on MOST.

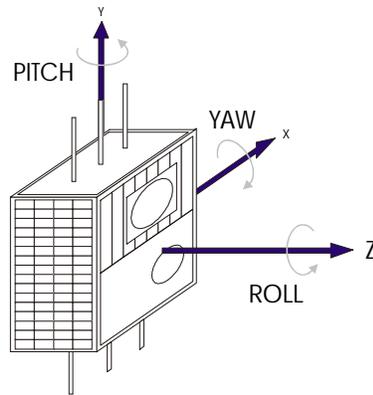


Figure 2.3: Euler Angle Rotations

2.2.1 Introduction to Quaternions

Quaternions are an extension of complex numbers into a four dimensional space. A complex number has one real part and one imaginary part.

$$(x + iy) \tag{2.1}$$

$$i * i = -1$$

A quaternion has one real part and three distinct imaginary parts corresponding to three distinct roots of -1 :

$$(w + ix + jy + kz) \tag{2.2}$$

$$i * i = -1 \quad i * j = -j * i = k$$

$$j * j = -1 \quad j * k = -k * j = i$$

$$k * k = -1 \quad k * i = -i * k = j$$

Quaternions do not exhibit the “gimbal lock” singularity present in the Euler angle representation.

Some important properties of quaternions of significance for this project are listed. Quaternions are not commutative, so the order of multiplication must be preserved.

$$\bar{\mathbf{q}}_1 \otimes \bar{\mathbf{q}}_2 \neq \bar{\mathbf{q}}_2 \otimes \bar{\mathbf{q}}_1 \quad (2.3)$$

Quaternion multiplication can be written as matrix multiplication in much the same way as the cross product of two three vectors.

$$\bar{\mathbf{q}}_1 \otimes \bar{\mathbf{q}}_2 = [\bar{\mathbf{q}}_1] \bar{\mathbf{q}}_2 = \{\bar{\mathbf{q}}_2\} \bar{\mathbf{q}}_1 \quad (2.4)$$

where

$$[\bar{\mathbf{q}}_1] = \begin{bmatrix} w_1 & z_1 & -y_1 & x_1 \\ -z_1 & w_1 & x_1 & y_1 \\ y_1 & -x_1 & w_1 & z_1 \\ -x_1 & -y_1 & -z_1 & w_1 \end{bmatrix} \quad \{\bar{\mathbf{q}}_2\} = \begin{bmatrix} w_2 & -z_2 & y_2 & x_2 \\ z_2 & w_2 & -x_2 & y_2 \\ -y_2 & x_2 & w_2 & z_2 \\ -x_2 & -y_2 & -z_2 & w_2 \end{bmatrix} \quad (2.5)$$

Quaternions have many other useful properties which will not be detailed here.

2.2.2 Quaternions as Rotations

Any arbitrary series of Euler rotations can be reduced to a single rotation about an axis, sometimes called the screw axis. Quaternion based rotations make use of this fact. The attitude quaternion is defined as:

$$\bar{\mathbf{q}} = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix} = w + \mathbf{i}x + \mathbf{j}y + \mathbf{k}z \quad (2.6)$$

$$\mathbf{q} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad q_4 = w \quad (2.7)$$

where

$$\mathbf{q} = \hat{\mathbf{n}} \sin(\theta/2) \quad q_4 = \cos(\theta/2) \quad (2.8)$$

and $\hat{\mathbf{n}}$ is a unit vector in the direction of the screw axis.

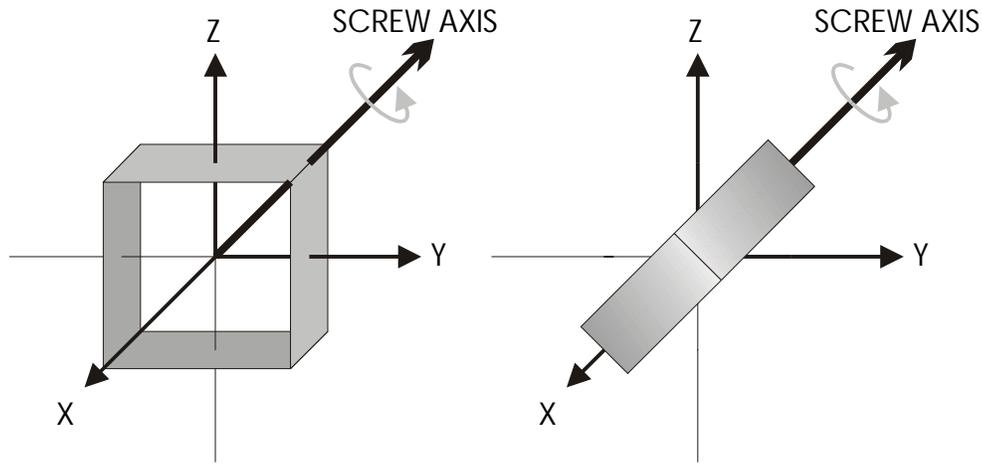


Figure 2.4: 90° Rotation about Screw Axis

Given the quaternion attitude representation $\bar{\mathbf{a}}$. To rotate it with a quaternion $\bar{\mathbf{q}}$ you must left multiply and right multiply as follows.

$$\bar{\mathbf{a}}_{rotated} = \bar{\mathbf{q}} \otimes \bar{\mathbf{a}} \otimes \bar{\mathbf{q}}^{-1} \quad (2.9)$$

It is also possible to write a rotation quaternion as a rotation matrix as follows:

$$\mathbf{R}(\bar{\mathbf{q}}) = \begin{bmatrix} x^2 - y^2 - z^2 + w^2 & 2(xy + zw) & 2(xz - yw) \\ 2(xy - zw) & -x^2 + y^2 - z^2 + w^2 & 2(yz + xw) \\ 2(xz + yw) & 2(yz - xw) & -x^2 - y^2 + z^2 + w^2 \end{bmatrix} \quad (2.10)$$

Finally, it is useful to define $\bar{\boldsymbol{\omega}}$ here, where $\boldsymbol{\omega}$ is the angular velocity vector:

$$\bar{\boldsymbol{\omega}} = \begin{bmatrix} \boldsymbol{\omega} \\ 0 \end{bmatrix} \quad \boldsymbol{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (2.11)$$

2.3 Equations of Motion

The equations of motion for the attitude of a rigid body are

$$\frac{d}{dt} {}^c \bar{\mathbf{q}}_t = \frac{1}{2} [{}^c \bar{\boldsymbol{\omega}}_{cl}] {}^c \bar{\mathbf{q}}_t \quad (2.12)$$

$${}^c \mathbf{g} = \hat{\mathbf{I}} \frac{d}{dt} {}^c \boldsymbol{\omega}_{cl} + {}^c \boldsymbol{\omega}_{cl} \times \hat{\mathbf{I}} {}^c \boldsymbol{\omega}_{cl} \quad (2.13)$$

The torque ${}^c \mathbf{g}$ is the total torque on the system. It includes all control torques and

disturbance torques. These equations will be modified in several ways for the purposes of this project. See Section 6.2.

2.4 Numerical Methods

The nonlinear equations of motion were solved using the fourth order Runge-Kutta method. See [5] for details.

Given:

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}, t) \quad \text{and} \quad \mathbf{x}_o, f_o \quad (2.14)$$

the solution is given by:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{1}{6} \Delta t (W_{k1} + 2W_{k2} + 2W_{k3} + W_{k4}) \quad (2.15)$$

where:

$$\begin{aligned} W_{k1} &= f_k \\ W_{k2} &= f\left(\mathbf{x}_k + \frac{\Delta t W_{k1}}{2}, t + \frac{\Delta t}{2}\right) \\ W_{k3} &= f\left(\mathbf{x}_k + \frac{\Delta t W_{k2}}{2}, t + \frac{\Delta t}{2}\right) \\ W_{k4} &= f(\mathbf{x}_k + \Delta t W_{k3}, t + \Delta t) \end{aligned} \quad (2.16)$$

In practice some investigation should be done to insure that this solution converges for the time step used.

3. State Estimation

One method for estimating the attitude of a dynamic system is the use of a state observer from control systems theory. The discrete time state observer will be developed here. The Kalman filter based observer gain will be presented as the optimal solution to the state estimation problem. A practical algorithm for implementing the Kalman filter is shown. Finally, the derivation is duplicated for a non-linear system.

3.1 State Model

The general state model for a non-linear system is given by:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= f(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{w}(t) \\ \mathbf{y}(t) &= g(\mathbf{x}(t), t) + \mathbf{v}(t)\end{aligned}\tag{3.1}$$

where

- $\mathbf{x}(t)$ – the state vector,
- $\mathbf{u}(t)$ – is the vector formed by inputs to actuators,
- $\mathbf{y}(t)$ – is the vector formed by outputs from sensors,
- $\mathbf{w}(t)$ – is the process noise, and
- $\mathbf{v}(t)$ – is the sensor noise.

It is important to note that this is a continuous time model. These equations were solved using numerical methods discussed in Section 2.4. With numerical methods, the continuous time interpretation used is essentially interchangeable with a discrete time interpretation.

The initial development of observer theory and the Kalman filter are presented using the following linear time varying state model.

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k \\ \mathbf{y}_k &= \mathbf{C}_k \mathbf{x}_k + \mathbf{D}_k \mathbf{u}_k + \mathbf{v}_k\end{aligned}\tag{3.2}$$

3.2 Observers

In order to control the attitude of the satellite, an estimate of the state, \mathbf{x}_k , is desirable. The state estimate will be denoted by $\hat{\mathbf{x}}_k$. An estimate of the state could be formed by direct computer simulation of equation (3.2). The abilities of this method to accurately predict the state would be contingent on knowledge of the initial condition, \mathbf{x}_0 . The initial condition,

however, is not generally known. An estimate can be formed by guessing \mathbf{x}_0 , and then correcting with measurements from the actual system. Methods for implementing corrections based on sensor measurements are called observers. Two observers useful in understanding the Kalman Filter are presented below. The derivation follows [4] and [9].

3.2.1 Identity Observer

The full state identity observer uses the measurement of \mathbf{y}_k , and all earlier measurements to predict $\hat{\mathbf{x}}_{k+1}$.

$$\begin{aligned}\hat{\mathbf{x}}_{k+1} &= \mathbf{A}_k \hat{\mathbf{x}}_k + \mathbf{B}_k \mathbf{u}_k + \mathbf{K}_k (\hat{\mathbf{y}}_k - \mathbf{y}_k) \\ \hat{\mathbf{y}}_k &= \mathbf{C}_k \hat{\mathbf{x}}_k + \mathbf{D}_k \mathbf{u}_k\end{aligned}\quad (3.3)$$

It can be shown that the state estimation error $\mathbf{e}_k = \hat{\mathbf{x}}_k - \mathbf{x}_k$ obeys the difference equation:

$$\mathbf{e}_{k+1} = (\mathbf{A}_k - \mathbf{K}_k \mathbf{C}_k) \mathbf{e}_k \quad (3.4)$$

If the eigenvalues of $\mathbf{A}_k - \mathbf{K}_k \mathbf{C}_k$ are stable, the estimation error will asymptotically approach zero.

3.2.2 Current Estimator

If the computation time to generate $\hat{\mathbf{x}}_k$ is short compared to the sample period, the identity observer can be improved by making use of \mathbf{y}_{k+1} in the estimate of $\hat{\mathbf{x}}_{k+1}$. An *a priori* estimate, $\hat{\mathbf{x}}_k^-$, is formulated:

$$\hat{\mathbf{x}}_k^- = \mathbf{A}_{k-1} \hat{\mathbf{x}}_{k-1} + \mathbf{B}_{k-1} \mathbf{u}_{k-1} \quad (3.5)$$

Then, the estimate is corrected when the measurement is made to form an *a posteriori* estimate, $\hat{\mathbf{x}}_k$.

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{C}_k \hat{\mathbf{x}}_k^- - \mathbf{y}_k) \quad (3.6)$$

This observer is known as the current estimator. It can be shown that the estimate error formed with (3.5) & (3.6) obeys the difference equation.

$$\mathbf{e}_{k+1} = (\mathbf{A}_k - \mathbf{K}_k \mathbf{C}_k \mathbf{A}_k) \mathbf{e}_k \quad (3.7)$$

If the eigenvalues of $\mathbf{A}_k - \mathbf{K}_k \mathbf{C}_k \mathbf{A}_k$ are stable, the estimation error will asymptotically approach zero.

3.3 Optimal Estimation

The Kalman Filter is a solution to an optimal estimation problem. The observer gain can be chosen to minimize the estimate error covariance

$$\mathbf{P}_k = E[(\hat{\mathbf{x}}_k - \mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T] \quad (3.8)$$

in the presence of process and sensor noise, \mathbf{w}_k and \mathbf{v}_k . If the noises are independent multivariate normal distributions with zero mean and covariance matrices, \mathbf{Q}_k and \mathbf{R}_k , respectively then the solution to the minimization is the Kalman filter.

$$\begin{aligned} p(\mathbf{w}) &\sim N(\mathbf{0}, \mathbf{Q}_k) \\ p(\mathbf{v}) &\sim N(\mathbf{0}, \mathbf{R}_k) \end{aligned} \quad (3.9)$$

The optimal observer gain is given by

$$\mathbf{K}_k = \mathbf{P}_k \mathbf{C}_k^T (\mathbf{C}_k \mathbf{P}_k \mathbf{C}_k^T + \mathbf{R}_k)^{-1} \quad (3.10)$$

where \mathbf{P}_k is the solution to the discrete time algebraic Riccati equation.

$$\mathbf{P}_{k+1} = \mathbf{A}_k \left[\mathbf{P}_k - \mathbf{P}_k \mathbf{C}_k^T (\mathbf{C}_k \mathbf{P}_k \mathbf{C}_k^T + \mathbf{R}_k)^{-1} \mathbf{C}_k \mathbf{P}_k \right] \mathbf{A}_k^T + \mathbf{Q}_k \quad (3.11)$$

The Kalman filter is only optimal if the conditions stated are met, and in practical application noises do not have normal distributions. The Kalman filters optimality is not the most interesting aspect of its function however.

Consider that the state covariance matrix can be viewed as a measure of the uncertainty of the state model, and the sensor covariance matrix can be viewed as a measure of the quality of the measurement. In this way the Kalman filter has important implications. The sensor covariance matrix weighs the use of each sensor based on its individual accuracy. If a sensor becomes unreliable due to a known condition, the sensor covariance could be changed to reduce that sensor's role in the state estimate. The state covariance matrix weighs the use of the *a priori* estimate in the *a posteriori* estimate, so if the model is very uncertain then the sensors can be emphasized over the prediction. The issue of sensor

weighting will be discussed further in Section 4. Clearly, the Kalman filter is very useful when dealing with ‘real’ sensor performance and practical state models.

3.3.1 Practical Kalman Filter Algorithm

In practice, the optimal estimator discussed above is implemented by successive time and measurement updates. First, a time update is performed to predict the *a priori* state estimate and state error covariance. Then, a measurement update is performed to form the corrected *a posteriori* state estimate and state error covariance. This sequence of time and measurement update is then repeated for successive time intervals.

The time update equations are:

$$\text{(compute } a \text{ priori state estimate)} \quad \hat{\mathbf{x}}_k^- = \mathbf{A}_{k-1} \hat{\mathbf{x}}_{k-1} + \mathbf{B}_{k-1} \mathbf{u}_{k-1} \quad (3.12)$$

$$\text{(compute } a \text{ priori error covariance)} \quad \mathbf{P}_k^- = \mathbf{A}_{k-1} \mathbf{P}_{k-1} \mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1} \quad (3.13)$$

The measurement update equations are:

$$\text{(compute observer gain)} \quad \mathbf{K}_k = \mathbf{P}_k^- \mathbf{C}_k^T (\mathbf{C}_k \mathbf{P}_k^- \mathbf{C}_k^T + \mathbf{R}_k)^{-1} \quad (3.14)$$

$$\text{(compute } a \text{ posteriori estimate)} \quad \hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{C}_k \hat{\mathbf{x}}_k^-) \quad (3.15)$$

$$\text{(compute } a \text{ posteriori error covariance)} \quad \mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k) \mathbf{P}_k^- \quad (3.16)$$

3.3.2 Extended Kalman Filter

The Extended Kalman Filter makes use of the same basic observer theory already stated and applies it to non-linear systems of the form:

$$\begin{aligned} \mathbf{x}_{k+1} &= f(\mathbf{x}_k, \mathbf{u}_k, k) + \mathbf{w}_k \\ \mathbf{y}_k &= g(\mathbf{x}_k, k) + \mathbf{v}_k \end{aligned} \quad (3.17)$$

Cycle by cycle, these equations can be linearized about the current *a priori* state estimate.

$$\begin{aligned}
\mathbf{A}_k &= \left\{ A_{i,j}(k) = \frac{\partial}{\partial x_j} f_i(\mathbf{x}_k, \mathbf{u}_k, k) \Big|_{\hat{\mathbf{x}}_k^-, \mathbf{u}_k, k} \quad i = 1..n, j = 1..n \right\} \\
\mathbf{B}_k &= \left\{ B_{i,j}(k) = \frac{\partial}{\partial u_j} f_i(\mathbf{x}_k, \mathbf{u}_k, k) \Big|_{\hat{\mathbf{x}}_k^-, \mathbf{u}_k, k} \quad i = 1..n, j = 1..m \right\} \\
\mathbf{C}_k &= \left\{ C_{i,j}(k) = \frac{\partial}{\partial x_j} g_i(\mathbf{x}_k, \mathbf{u}_k, k) \Big|_{\hat{\mathbf{x}}_k^-, \mathbf{u}_k, k} \quad i = 1..\ell, j = 1..n \right\}
\end{aligned} \tag{3.18}$$

Now, we can use these matrices in the calculation of error covariance.

EKF time update equations are:

$$\text{(compute } a \text{ priori state estimate)} \quad \hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, k-1) \tag{3.19}$$

$$\text{(compute } a \text{ priori error covariance)} \quad \mathbf{P}_k^- = \mathbf{A}_{k-1} \mathbf{P}_{k-1} \mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1} \tag{3.20}$$

EKF measurement update equations are:

$$\text{(compute observer gain)} \quad \mathbf{K}_k = \mathbf{P}_k^- \mathbf{C}_k^T (\mathbf{C}_k \mathbf{P}_k^- \mathbf{C}_k^T + \mathbf{R}_k)^{-1} \tag{3.21}$$

$$\text{(compute } a \text{ posteriori estimate)} \quad \hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - g(\hat{\mathbf{x}}_k^-, k)) \tag{3.22}$$

$$\text{(compute } a \text{ posteriori error covariance)} \quad \mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k) \mathbf{P}_k^- \tag{3.33}$$

Along with the properties of the Kalman Filter already mentioned, there are two interesting additional implications for the Extended Kalman Filter. All of the state estimate propagation equations rely on the non-linear state models. In this way, the filter does not imply linearization of the attitude control system in general. Finally, the gain matrix, \mathbf{K} , is dependent on the linearized version of $g(\mathbf{x}_k, k)$. This has the effect of amplifying the sensors response curve in the region that is actually affecting the measurements at that instant.

4. Sensor Models

The satellite will have four types of sensors for the purpose of attitude determination: rate sensors, magnetometers, sun sensors and a star tracker. The sensors will be described briefly here. Details on the magnetometers can be found in Section 5.

4.1 Rate Sensors

There will be at least three rate sensors mounted in a linearly independent fashion. More rate sensors can improve the quality of the data provided but also increase cost and complexity.

4.2 Magnetometers

Details on magnetometers are discussed in Section 5.

4.3 Sun Sensors

Sun sensors measure the angle of the sun relative to the plane in which the sensor is mounted. Depending on the number of sensors and their orientation on the satellite, several restrictions may apply. A sun sensor needs to be in the sun's bore sight to within its sensitivity angle, for example $\pm 30^\circ$, in order to provide useful data. While a particular sensor is not pointed at the sun, that sensor needs to be ignored so that it does not interfere with the attitude determination.

4.4 Star Tracker

Star trackers can operate in several modes. A differential star tracker measures the movement of fixed points of interest in its field of view. Other star trackers compare an image with a reference star field.

One especially interesting aspect of star tracker operation is that it will not be able to provide readings at the same rate as the other sensors. The star tracker will operate at 1 Hz while the other sensors may operate as fast as 10Hz.

5. Magnetometers

Magnetometers are particularly convenient sensors for satellite attitude determination. They are relatively inexpensive and can give information about the satellite's absolute attitude. Unlike gyros, magnetometers have no drift. For this project, it will be assumed that the magnetometers generate a vector representing the magnetic field measured along each axes of the Control CS. Using the IGRF model described below, the Earth's magnetic field vector at the satellite's known location can be determined. Finally, the measurement can be compared with the Earth's magnetic field. Using observer based methods described in Section 3, the correct attitude for the spacecraft can be found.

5.1 IGRF Model

The International Geomagnetic Reference Field (IGRF) model is a series of coefficients of the spherical harmonics of the Earth's magnetic field. They can be used to determine the Earth's magnetic field to good precision at any position and time. A C program, 'IGRFgeodetic.c', [13], can take satellite position in Altitude-Latitude-Longitude coordinates and produce the vector of the Earth's magnetic field at that position in North-East-Down coordinates.

5.1.1 Altitude-Latitude-Longitude Coordinate System (ALL CS)

The ALL CS is an Earth fixed spherical coordinates system. Latitude and Longitude are defined in the traditional way used in cartography. Altitude is defined to be the distance from the Earth's surface for a spherical Earth.

5.1.2 North-East-Down Coordinate System (NED CS)

As an observer standing on the Earth, one can consider the environment to be flat and the directions North, East and Down equivalent to the x-axis, y-axis and negative z-axis in the classical Cartesian coordinate system. This approximation is of little use for distances greater than a few hundred kilometers. Unfortunately, this is the traditional way to view the Earth's magnetic field.

It is a relatively simple matter to transform the NED CS into the Orbit CS. The ‘Down’ direction in the NED CS is equivalent to the z-axis of the Orbit CS. Thus, the NED CS must be equal to the Orbit CS to within a rotation about the z-axis of the Orbit CS. The angle of that rotation is given by:

$$\theta_z = a \times (|inclination| + |latitude|), \quad a = \begin{cases} -1 & \text{if ascending} \\ +1 & \text{if descending} \end{cases} \quad (5.1)$$

where:

<i>inclination</i>	is one of the classical orbital elements of the satellite,
<i>latitude</i>	is the latitude dimension of the ALL CS,
<i>ascending</i>	is true while the satellite is traveling from the south pole to the north pole, or while the <i>latitude</i> is increasing,
<i>descending</i>	is true while the satellite is traveling from the north pole to the south pole, or while the <i>latitude</i> is decreasing.

The matlab script, ‘IGRFgeodetic.m’, passes ALL CS coordinates to the program ‘IGRFgeodetic.c’ and converts the returned values to the Orbit CS. Note that the *inclination* variable, set for UO-22 in ‘IGRFgeodetic.m’, can be easily adjusted.

5.2 State Estimation Based on Vector Measurements

One single vector measurement contains information of two of the three attitude degrees of freedom only. One way to determine full attitude from magnetic field measurements is to use the measurements in an observer as described in Section 3. To form an observer based on vector measurements, we must define a measurement estimate, an innovation quantity, and a state estimate update formula.

Given the satellite’s position in orbit and the IGRF model discussed above, it is possible to know the Earth’s magnetic field vector in the Orbit CS. Using the a priori attitude estimate, the magnetic field vector can be expressed in the Control CS. The measurement estimate is thus available although not quite in the classical form of $g(\hat{\mathbf{x}}_k^-, k)$ in (3.17).

$$\begin{aligned}
g(\hat{\mathbf{x}}_k^-, k) &= g({}^C \bar{\mathbf{q}}_0, \text{satellite position, IGRF model}) \\
&= R({}^C \bar{\mathbf{q}}_0)^o \mathbf{b}_{IGRF} \\
&= \hat{\mathbf{y}}
\end{aligned} \tag{5.2}$$

Next, an innovation quantity must be defined based on the measurement estimate, $\hat{\mathbf{y}}$, and the actual magnetometer measurement, \mathbf{y} .

Typically, the innovation is defined to be the difference between estimated and actual measurement values, as seen in (3.3). However, in the case of vector measurements of attitude, the difference operation is not appropriate. The cross product is used to form the innovation instead, see [2]. A modification of that definition is described below.

The result of $\hat{\mathbf{y}} \times \mathbf{y}$ is a vector in the direction of the axis of rotation required to align $\hat{\mathbf{y}}$ with \mathbf{y} . Also, if $\hat{\mathbf{y}}$ and \mathbf{y} have unit magnitude, the magnitude of $\hat{\mathbf{y}} \times \mathbf{y}$ is the sine of the angle between them. The quaternion based rotation required to align the estimate with the measurement is given by $\bar{\mathbf{v}}$ as follows:

$$\begin{aligned}
\mathbf{v} &= \frac{\hat{\mathbf{y}} \times \mathbf{y}}{\|\hat{\mathbf{y}}\| \cdot \|\mathbf{y}\|} \\
\theta &= \sin^{-1} \|\mathbf{v}\| \\
\bar{\mathbf{v}} &= \begin{bmatrix} \mathbf{v} \cdot \frac{\sin(\theta/2)}{\|\mathbf{v}\|} \\ \cos(\theta/2) \end{bmatrix}
\end{aligned} \tag{5.3}$$

The value $\|\mathbf{v}\|$ will frequently be very close to zero. Therefore, the above definition for $\bar{\mathbf{v}}$, while mathematically precise, is not numerically stable. Since it is not necessary to recover the precise value θ , the following approximation will be used instead.

$$\begin{aligned}
\mathbf{v} &= \frac{\hat{\mathbf{y}} \times \mathbf{y}}{\|\hat{\mathbf{y}}\| \cdot \|\mathbf{y}\|} \\
\bar{\mathbf{v}} &= \begin{bmatrix} \mathbf{v} \\ \sqrt{1 - \|\mathbf{v}\|^2} \end{bmatrix}
\end{aligned} \tag{5.4}$$

The rotation $\bar{\mathbf{v}}$ will be considered the innovation of the observer.

Instead of the standard state estimate update formula, rotation of the state estimate will be used.

$$\hat{\mathbf{x}}^+ = \begin{bmatrix} \hat{\mathbf{q}}^+ \\ \hat{\boldsymbol{\omega}}^- \end{bmatrix}, \quad \hat{\mathbf{q}}^+ = \begin{bmatrix} R(\mathbf{K}_q \bar{\mathbf{v}}) \hat{\mathbf{q}}^- \\ \sqrt{1 - |R(\mathbf{K}_q \bar{\mathbf{v}}) \hat{\mathbf{q}}^-|^2} \end{bmatrix} \quad (5.5)$$

\mathbf{K}_q is the observer gain and can be specified by way of a Kalman filter. In [2] $\hat{\boldsymbol{\omega}}$ was updated using a conventional definition of innovation. No update of $\hat{\boldsymbol{\omega}}$ was used here because vector measurements of magnetic field do not directly affect the rates. However, due to the rotating coordinate systems an update of $\hat{\boldsymbol{\omega}}$ may be required. This area was not investigated in detail. It should be considered carefully in any future work.

5.2.1 Kalman Filtering Based on Vector Measurements

To form a Kalman filter from the observer above, the linearization outlined in Section 3 must be carried out, and covariance and gain update formulas must be defined. This process was not completed due to time constraints, but a direct application of the Kalman filter formulas in Section 3 should yield good results. The Kalman update formulas from [2] were attempted, see Section 7.5.

6. UoSAT-OSCAR 22 (UO-22)

In order to evaluate the attitude estimation algorithm developed in Section 5, real satellite data needed to be used. UO-22 was selected as the test subject. UO-22 is a Radio Amateur Satellite launched by the University of Surrey in 1991. It has available magnetometer measurements, known physical attributes and a known orbit. It is in a low earth orbit with a 98° inclination, very much like MOST. This is ideally suited to magnetometer based attitude determination. Also, its attitude is known to within a single rotation, about its spinning axis, because it is gravity gradient and single-spin stabilized.

6.1 UO-22 Telemetry & Orbital Elements

The Space Flight Laboratory had easy access to telemetry data and orbital elements from UO-22 because of previous work [3]. Whole orbit telemetry data files and ground station reports of orbital elements from June to December 1999 were available. The program ‘vbtlm.exe’ was used to decode the telemetry files and output the data as ASCII text.

6.1.1 Magnetometer Readings

‘vbtlm.exe’ was used to decode magnetometer readings for the time window: 25 Nov 1999 02:26:35 to 25 Nov 1999 23:59:35. The magnetometer telemetry data was in the form of voltage readings taken from the instruments. No information about the transformation from voltage level to magnetic field intensity was available. By looking at how magnetometer readings were encoded for UoSAT-OSCAR 11 (UO-11), a similar satellite built by the same organization, it was possible to make some assumptions about UO-22’s magnetometers. According to [8], UO-11’s magnetometer voltage readings were decoded by the following functions:

$$\begin{aligned} \text{Nav mag } X \text{ axis: } H &= (0.1485N - 68)\mu T \\ \text{Nav mag } Y \text{ axis: } H &= (0.1523N - 69.3)\mu T \\ \text{Nav mag } Z \text{ axis: } H &= (0.1507N - 69)\mu T \end{aligned} \tag{6.1}$$

where N is the voltage read from the relevant magnetometer. An assumption was made that UO-22’s magnetometer readings must be of a similar nature. Thus, it was assumed that the

magnetometers on UO-22 are linear with a constant offset.

The offset for the x-axis and y-axis magnetometer measurements on UO-22 could be calculated. The satellite rotates about its z-axis once every ten minutes. Due to the rotation about the z-axis, the x-axis and y-axis vector components change cyclically between positive and negative. During ten minutes of satellite motion, the Earth's magnetic field vector changes very little. Therefore, the mean of the x-axis and y-axis components must be zero. Hence, the average reading of the x-axis and y-axis magnetometer should be equal to the offset value, and the offset is removed by subtracting the average from the measurement. The same idea was used to eliminate the offset term from the z-axis measurement although the justification is less direct. The formulas for UO-11's magnetometers show that offset values for different axes are almost equal. The calculated offset values for UO-22's magnetometers were almost equal as well.

Only the direction of the measured magnetic field vector is required. Thus, it is only necessary to consider the relative magnitude of the three sensor readings. The constants of proportionality for the magnetometers on UO-11 were 0.1485, 0.1523, 0.1507. While these are not exactly equal, the difference is small. This motivated the assumption that, for UO-22, the difference could be ignored. With the assumption of a fixed constant of proportionality, no attempt to discover the value of the constant is necessary.

To process the telemetry data from UO-22, the three axis magnetometer measurements were read from a text file, the average reading subtracted, and the magnitude normalized. This was done by the function 'getbmag.m'. Thus, a unit vector in the direction of the measured magnetic field vector was available.

6.1.2 Orbit Propagation

Using Satellite Tool Kit it was possible to use the available orbital elements to project UO-22's orbit during the time window of interest, defined in 6.1.1. A report containing a representation of the satellites position above the Earth in x-y-z Earth fixed coordinates was generated from Satellite Tool Kit. This data was converted to the ALL CS for use by

‘IGRFgeodetic.m’ using the following formula.

$$\begin{aligned} R &= (x^2 + y^2 + z^2)^{1/2} \\ Lat &= \frac{180^\circ}{\pi} \sin^{-1}\left(\frac{z}{R}\right) \\ Lon &= \frac{180^\circ}{\pi} \tan^{-1}\left(\frac{y}{x}\right) \\ Alt &= R - 6371.2\text{km} \end{aligned} \tag{6.2}$$

6.2 Attitude Kinematics & Dynamics of UO-22

Some basic changes to the coordinate systems and equations of motion to be used must be made to accommodate UO-22.

6.2.1 Control Coordinate System (Control CS)

The Control CS is specified differently for UO-22. It is a right orthogonal coordinate system. The origin is placed at the center of mass of the satellite and the axes are aligned with the principal axis of the satellite’s moment of inertia. For UO-22 the z-axis is chosen to point approximately along the gravity gradient boom, which is the axis of least moment of inertia.

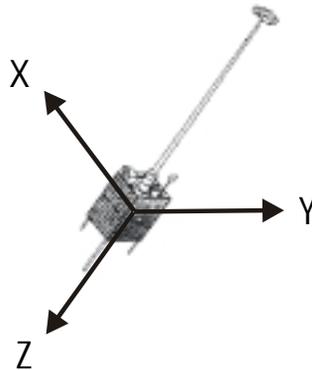


Figure 5: Control Coordinate System - UO-22

6.2.2 Gravity Gradient Effect

The gravity gradient effect describes the effect of minor differences in the force of gravity at different points in a satellite. An orbiting object tends to align its axis of least moment of

inertia with the gravitational field. For instance, a barbell-shaped satellite like UO-22 would tend to align its z-axis with the gravitational field. If the z-axis were parallel to the gravitational field, then the tips of the satellite would be equal distance from the earth. The forces on the satellite in the direction of the gravitational center of mass would be balanced. This is an unstable equilibrium however. Any small disturbance from that orientation would put one end closer than the other. There would be greater force on the closer end causing an unbalanced torque. While the orbit would not be affected overall, the unbalanced torque would tend to align the axis of least moment of inertia with the gravitational field.

The gravity gradient effect can be modeled mathematically [10] as

$${}^c \mathbf{g}_{gg} = 3\omega_o^2 \left({}^c \mathbf{k}_o \times \hat{\mathbf{I}} {}^c \mathbf{k}_o \right) \quad (6.3)$$

with

$${}^c \mathbf{k}_o = \mathbf{R} \left({}^c \bar{\mathbf{q}}_o \right) \times {}^o \mathbf{k}_o \quad (6.4)$$

Where ω_o is the orbital rate, 1.05mrad/min, of the satellite and ${}^o \mathbf{k}_o$ is the z-axis unit vector.

6.2.3 Modified Equations of Motion

For UO-22, the equations of motion were transformed from the standard set listed in Section 2.3.1. The torque balance equation was developed with the gravity gradient effect. This was done to better mimic the true dynamics of the satellite. All other torques on the system were neglected. The attitude equation was also changed. UO-22 is gravity gradient and spin stabilized, therefore, UO-22's attitude with respect to the Orbit CS is known and unchanging, except for a rotation about the z-axis. Due to this fact, it is much easier to understand UO-22's attitude over time as Control CS with respect to Orbit CS. The following are the modified equations used:

$$\frac{d}{dt} {}^c \boldsymbol{\omega}_{CI} = \hat{\mathbf{I}}^{-1} \left(-{}^c \boldsymbol{\omega}_{CI} \times \hat{\mathbf{I}}^c \boldsymbol{\omega}_{CI} + 3\omega_o^2 \left({}^c \mathbf{k}_o \times \hat{\mathbf{I}}^c \mathbf{k}_o \right) \right) \quad (6.5)$$

$$\frac{d}{dt} {}^c \bar{\mathbf{q}}_o = \frac{1}{2} \begin{bmatrix} {}^c \boldsymbol{\omega}_{CO} \\ 0 \end{bmatrix} \otimes {}^c \bar{\mathbf{q}}_o \quad (6.6)$$

$${}^c \boldsymbol{\omega}_{CO} = {}^c \boldsymbol{\omega}_{CI} - \mathbf{R}({}^c \bar{\mathbf{q}}_o) \begin{bmatrix} \omega_o \\ 0 \\ 0 \end{bmatrix} \quad (6.7)$$

6.2.4 Moment of Inertia

Several sources of physical information about the satellite were compared. The information gathered contained some inconsistency. Following is a listing of the sources found and the information provided:

From the Small Satellite Home Page at University of Surrey [16]:

...The 48.4kg microsatellite is box shaped with dimensions 350x350x650mm, with four solar panels on the larger facets, and a 6m gravity gradient boom...

From the AMSAT Home Page [15]:

...Weight: 48.4 kg
Size: 350 x 350 x 650 mm...

From the AMSAT Journal [14]:

...Height: 600 mm and 5 Meter Gravity Gradient Boo with 3 Kg tipmass
Width: 300 mm
Depth: 300 mm
Weight: 46 Kg...

The physical parameters used were: weight 48.4kg, size 350x350x650mm, boom length 6m, boom mass 3kg.

When looking at rotations about the x and y axes, UO-22 can be approximated with two

lumped point masses at the tips of a 6m mass-less rod. When looking at rotations about the z-axis, it can be approximated as a uniformly distributed mass over a 350mm x 350mm square.

The moment of inertia about the x and y axes is:

$$I_x = I_y = \sum_i \frac{1}{2} m_i r_i^2 \quad (6.8)$$

where r_i is the distance of m_i from the center of mass about the axis in question.

$$\begin{aligned} M_1 r_1 &= M_2 r_2 \\ r_1 &= 6m - r_2 \\ r_2 3kg &= (6m - r_2) 45.4kg \\ r_2 &= 5.6m \\ r_1 &= 0.4m \\ I_x, I_y &= \frac{1}{2} (3kg(5.6m)^2 + 45.4kg(0.4m)^2) \\ &= 50.7kg \cdot m^2 \end{aligned} \quad (6.9)$$

and the moment of inertia about the z-axis is simply:

$$\begin{aligned} I_z &= \frac{1}{2} 48.4kg(0.18m)^2 \\ &= 0.78kg \cdot m^2 \end{aligned} \quad (6.10)$$

7. Observer Experiments

Two types of simulation and one experiment were performed. The first simulation verified that the satellite dynamic model was operating correctly. The second simulation verified the observer design by using a dipole based magnetic field. The experiment tested the observer using real UO-22 magnetometer readings and IGRF based Earth's magnetic field predictions. This gave final verification that the full system was operational.

7.1 Dynamics Simulation

This simulation confirmed that the dynamic model presented in Section 6.2.3 and numerical integration method presented in Section 2.4 converged to the expected motion of the satellite. The matlab script used to run this simulation can be found in the 'matlab\igrf' directory and is called 'sat_sim.m'.

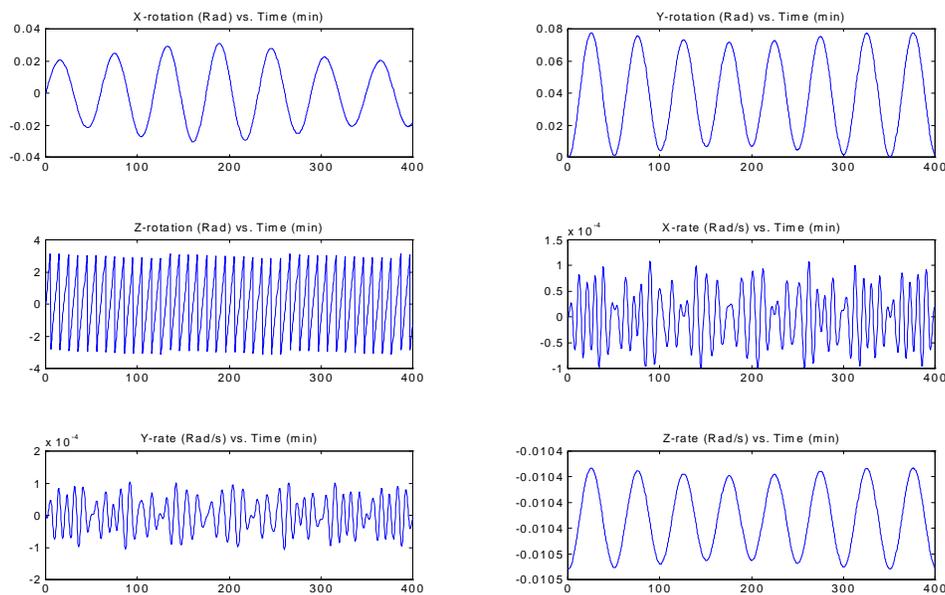


Figure 7.1: UO-22 Simulation

The first three subplots show the x-y-z Euler rotations of the Control CS with respect to the Orbit CS, or the angles associated with the rotation ${}^c\bar{\mathbf{q}}_O$. The other three subplots show the x-y-z angular rates of the Control CS with respect to the Orbit CS, expressed in the Control

CS, or the components of ${}^c \boldsymbol{\omega}_{CO}$. The coordinate systems and format of Figure 7.1 will be used for all results in this section. By looking at the z-rotation subplot, it is clear that the satellite is rotating rapidly in one direction about its z-axis. A small resolution problem can also be seen in this plot. The rotation seems to be modulated by a much lower frequency. This effect is occurred when the time step in the simulation was 30 seconds. A time step of 10 seconds reduces the effect significantly. The rotation in the z-rotation plot matches the spin of 0.0104rad/s imparted on the satellite by the initial conditions. The spin was added to resemble the true satellites spin stabilization of one rotation every ten minutes. By looking at the x-rotation and y-rotation subplots, it can be seen that the satellite is wobbling a very small amount, less than 0.1rad peak, about these axes. The wobble was due to the torque caused by the cross product of z-axis spinning and the orbital rate. The orbital rate is not visible in the Orbit CS; this was expected. The torque which caused the wobble was clearly opposed by a restoring torque which brought the satellite back toward zero rotation. The restoring torque is the gravity gradient effect discussed in Section 6.2.2. Finally, it is hard to decode information from the x-rate and y-rate subplots. These graphs do show the z-axis spinning discussed earlier, and they show that the rate about these axes is very small. These results match the expected motion and confirmed the methods used.

7.2 Observer Simulation (Dipole Model)

A dipole model, the magnetic field of a simple current loop, was generated, and a circular orbit was placed around it. A reference magnetic field vector was then determined based on that orbit. Two simulations of the satellite were then run in parallel: one was called the ‘real’ satellite; the other was called the ‘estimate’. The two simulations were given different initial conditions to represent the initial error of the observer. The magnetometer measurements were simulated by rotating the reference magnetic field vector with the ‘real’ satellite attitude. The predicted measurements were simulated by rotating the reference magnetic field vector with the ‘estimate’ satellite attitude.

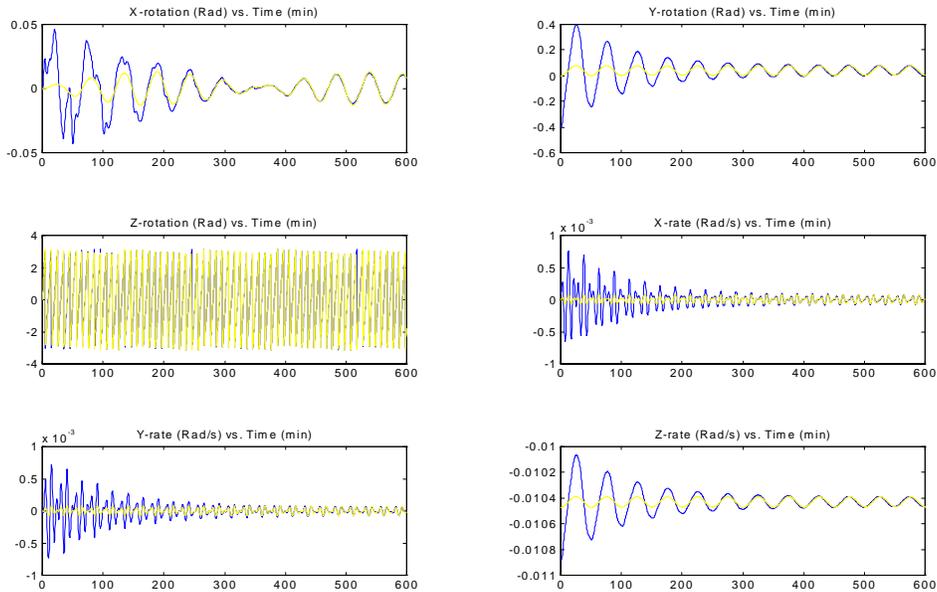


Figure 7.2: Observer Gain: 0.01 Initial Error: 6deg about y-axis

All graphs in this section show the ‘real’ satellite in yellow (or lighter) and the ‘estimate’ satellite in blue (or darker). The figure above shows the convergence of the estimate for a moderate observer gain and a moderate initial error about the y-axis. The observer could tolerate very high gain and very high initial error when the error was about the y-axis.

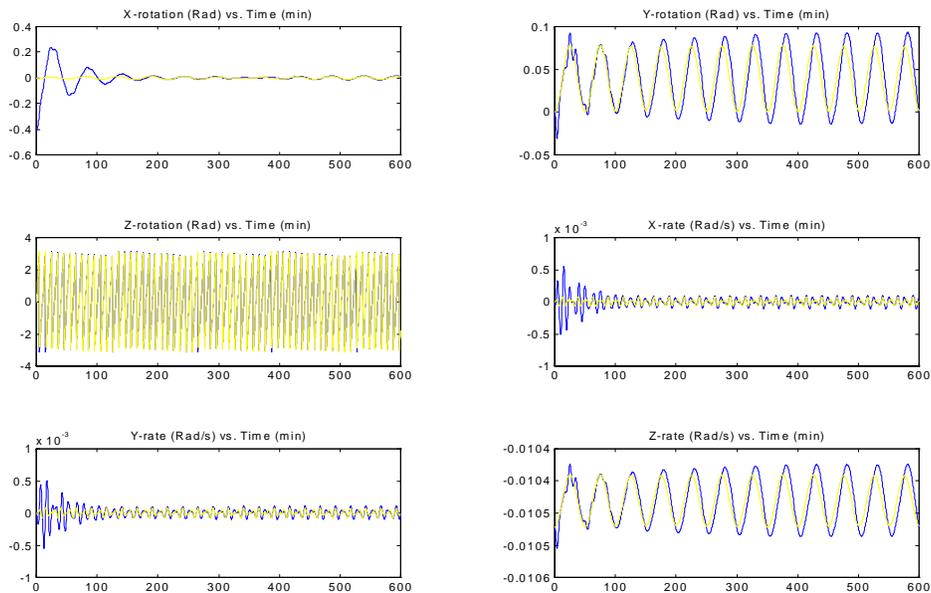


Figure 7.3: Observer Gain: 0.01 Initial Error: 0.2x

For the series of plots in Figure 7.3, the initial error was moved to the x-axis; the gain was unchanged. This series of plots is not converging on a good estimate of the state.

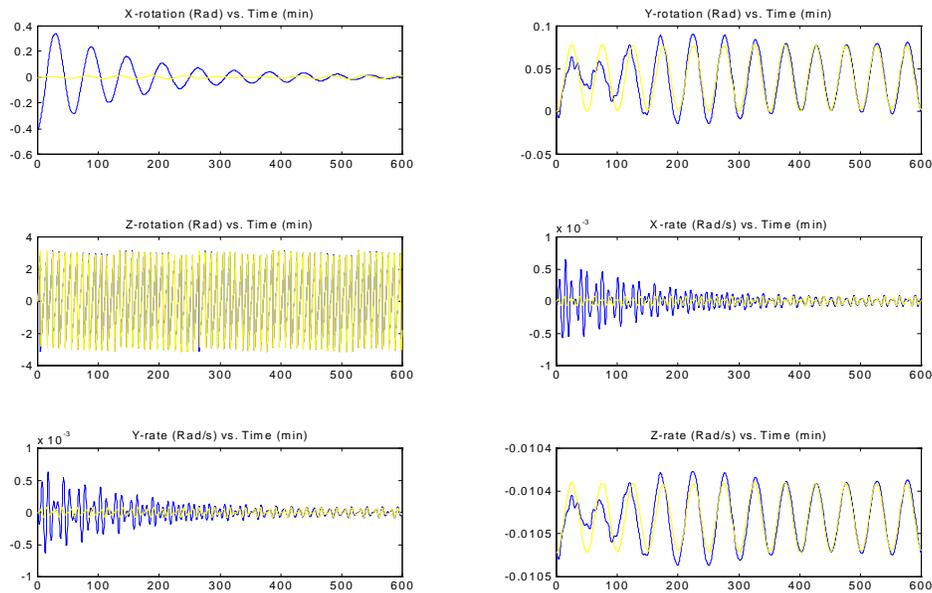


Figure 7.4: Gain=0.003 error=0.2x

The series of plots in Figure 7.4 represents a much smaller observer gain but maintains the initial error about the x-axis. This setup converged to a good estimate in a reasonable time. The observer could converge for somewhat higher initial errors as long as the gain was kept low.

The observer was able to converge for errors about the z-axis as well. As with x-axis errors, the gain had to be kept low. No simulations were done with very small rate errors; however, the observer was not able to converge for high rate errors. Clearly, the observer was capable of converging on a good state estimate. It is equally true that better performance than this should be possible. The observer was definitely more sensitive to errors about the x-axis and z-axis than to errors about the y-axis. This sensitivity was probably due to the lack of a rate update formula in the observer equations specified in Section 5.2.

7.3 Observer Experiment (Telemetry and IGRF model)

As described in Section 6.1, data for UO-22 was collected. Magnetometer readings and position information corresponding to each reading were taken. This data went through several stages of processing which have already been described. The experiments were performed exactly in the manner described for simulations in Section 7.2 except that there was no need for a ‘real’ satellite simulation. The measurement was taken directly from the magnetometers, and the measurement estimate was generated using the IGRF model.

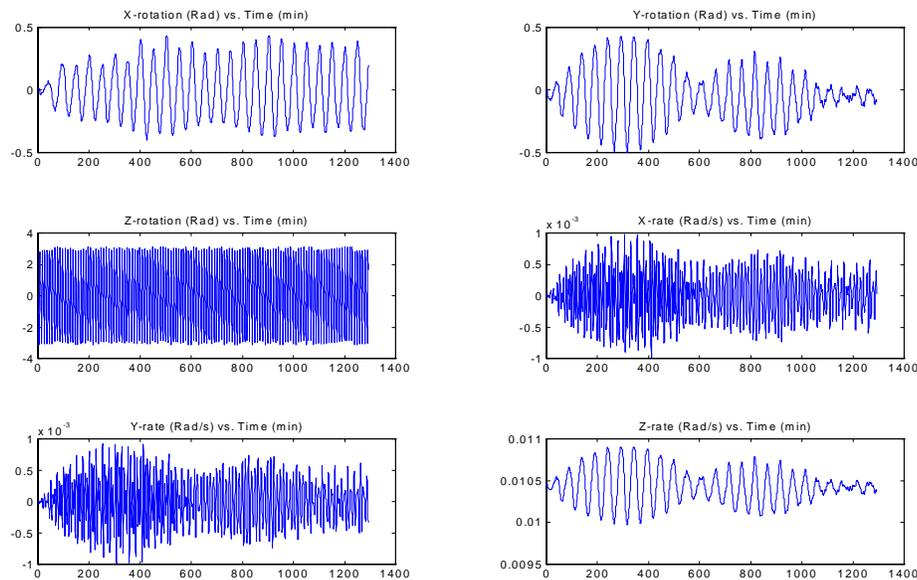


Figure 7.5: Observer Gain: 0.004 Initial Error: Unknown

The series of plots in Figure 7.5 follow the same basic format outlined in Section 7.1. The figure shows estimates. The real values of the six variables shown are not known, however, some assumptions can be made. The x-rotation and the y-rotation should be essentially zero, and they should certainly never perform a full or even quarter revolution. The z-rotation should display the characteristic 10min rotation period. The x-rate and y-rate should be essentially zero, and the z-rate should show the 10min rotation period as well. The figure shows the y-axis rotation converging on zero and the x-axis rotation teetering a small amount around zero. The z-axis rotation is displaying the appropriate spin, and the z-rate is converging to the appropriate spin rate. The other two rates are quit close to zero. There was a good deal of sensitivity to observer gain selection in this experiment. This was expected given the results in Section 7.2.

7.4 Discussion

There is an unexplained sensitivity to initial conditions about the x-axis and z-axis. There is also a definite relationship between the z-rate estimate and the y-rotation estimate. These affects appear to be related but the cause is unknown. The x-axis and y-axis should be relatively indistinguishable in the Control CS given the spin about the z-axis.

The problem probably lies in the lack of a rate update formula. In [2], a rate update formula is proposed. In [1], there is no need for a rate update formula because rates are not used in the equations of motion. Either of these approaches could yield improved results.

7.5 Other Simulations

An unsuccessful attempt was made to duplicate the innovation definition and Kalman update formulas from [2]. First, the innovation formulas were implemented in a fixed gain observer. The state estimate formulated in the simulations did not converge. However, the attitude kinematics and dynamics equations from [2] were not used directly. [2] used linearized equations but the simulations were performed with the nonlinear equations stated in Section 6.2.3. Also, [2] modeled some torques which were omitted in the simulations here. These differences may explain the poor results. Unfortunately, to investigate this problem would require more time.

After this initial work, a close examination of the innovation formula from [2] resulted in the innovation description in Section 5.2. With the new innovation formulas, the results presented in Section 7.1-7.4 were acquired. Some further attempts to integrate the results from [2] with the innovation definition given here were made without success.

8. Conclusions and Future Work

This project began with the goal of simulating attitude estimation techniques based on Kalman filtering for spacecraft attitude control. Many aspects have been completed, but there is still significant work to be done.

Spacecraft attitude kinematics and dynamics were discussed, including quaternion based attitude representations and numerical integration techniques. An investigation of estimation theory, especially the extended Kalman filter, was carried out. A concise description of estimation theory as it pertains to spacecraft attitude estimation was stated. The implications of a Kalman filter on various sensors was discussed. A detailed investigation of how to incorporate vector magnetic field measurements into a state estimation algorithm was shown. And a formula for observer innovation based on vector measurements was developed.

The satellite UO-22 was selected as a test subject to prove the results. Attitude kinematics and dynamics equations specific to UO-22 were formulated, and simulations were performed. An observer was simulated for both a dipole magnetic field and the IGRF model of the Earth's magnetic field. These simulations successfully captured the real dynamics of the satellite, and the state estimates converged to the satellites known attitude.

Significant work still needs to be done. A definition for innovation based on rate needs to be determined or ruled out completely. The performance of the observer demonstrated must be evaluated in detail; improvement may be possible. Finally, extended Kalman update formulas and cycle by cycle linearization of the equations of motion need to be developed. The matlab routines developed for this project include the facility to add all of these aspects.

Appendix A

Matlab Code Structure:

Dipole Model

Dipole Model

Observer Routine (sat_fix.m) - working

=====

Set the moment of inertia 'I' for later use by linear and non-linear satellite models.

Set sample rate in seconds 'T'

Set damping 'd' - always set to zero

set length of run in samples 'N'

simulate magnetic field vectors in orbital coordinates using dipole model

set initial conditions for state estimate 'x_est'

set initial conditions for the simulation 'x_real'

Set initial conditions for input 'g' - always set to zero

begin observer simulation

'q_est', 'w_est' and 'q_real' temporarily hold the current state estimate

predict the measurement 'b_pri'. The prediction is the current estimated attitude rotation applied to the simulated earth's magnetic field vector.

simulate the measurement 'b_meas'. The measurement is the current simulated attitude rotation applied to the simulated earth's magnetic field vector.

calculate the measurement innovation 'v'

update the attitude estimate 'q_est' by rotating about the 'v' axis with a fixed gain.

renormalize the attitude estimate

estimate the next cycle's a priori state

simulate the next cycle's actual state

generate output plot

Dipole Model (cont.)

Kalman Filter Routine (sat_ekf.m) - not working

=====

Set the moment of inertia 'I' for later use by linear and non-linear satellite models.

Set sample rate in seconds 'T'

Set damping 'd' - always set to zero

set length of run in samples 'N'

simulate magnetic field vectors in orbital coordinates using dipole model

set initial conditions for state estimate 'x_est'

set initial conditions for the simulation 'x_real'

Set initial conditions for input 'g' - always set to zero

initialize the error covariance matrixes

'Q' - Process error covariance

'R' - Measurement error covariance

'P' - Estimate error covariance

begin Kalman simulation

'q_est', 'w_est' and 'q_real' temporarily hold the current state estimate

predict the measurement 'b_pri'. The prediction is the current estimated attitude rotation applied to the simulated earth's magnetic field vector.

simulate the measurement 'b_meas'. The measurement is the current simulated attitude rotation applied to the simulated earth's magnetic field vector.

linearize about current state estimate and estimate the estimate error covariance 'P'

calculate the Kalman filter gain 'K'

update the state estimate covariance 'P'

calculate the measurement innovation 'v'

update the attitude estimate 'q_est' by rotating about the 'v' axis.

renormalize the attitude estimate

update the rate estimate 'w' in traditional manner

estimate the next cycle's a priori state

simulate the next cycle's actual state

generate output plot

Appendix B

Matlab Code:

Data Collection & IGRF Model

Data Collection & IGRF Model

Data Conversion - working
=====

This section describes the data conversion code and how it works.

First is the conversion to Earth's magnetic field vector for UO-22 at actual locations in the time window of interest.

Raw data was gathered from Satellite tool kit and processed using Excel. This data includes X- Y- Z- earth fixed position of UO-22 above the earth for the time window:

25Nov1999 02:26:35 to 25Nov1999 23:59:35

Output file - Uo22stk.txt

The data was then converted into Altitude Latitude and Longitude for use by 'IGRFgeodetic.m' which interfaces with 'IGRF geodetic.c' from Atsuhiko Sakura. creating Earth's Magnetic field vector in North East Down Earth fixed coordinates.

Output file - Uo22igrf.mat

The data was further converted to orbital coordinates using 'IGRFgeodetic.m' Once in orbital coordinates the data could be used by the Observer or Kalman filter routines.

Output file - Uo22be.mat

Now the conversion of UO-22 telemetry data to useful form.

Raw data was gathered from UO-22 telemetry files using Laura Hallidays program 'vbtlm' and processed with Excel. The result was - Uo22tel.txt

This data is normalized by the function 'getbmag.m'. 'getbmag.m' is called by the Observer or Kalman filter routines.

Data Collection & IGRF Model (cont.)

Observer Routine (sat_fix.m) - working

=====

Set the moment of inertia 'I' for later use by linear and non-linear satellite models.

Set sample rate in seconds 'T'

Set damping 'd' - always set to zero

load earths magnetic field vector 'b_IGRF'

load magnetometer data 'b_mag' via the function 'getbmag'
getbmag - reads and normalizes data file 'Uo22tel.txt'

Set initial conditions for state estimate 'x'

Set initial conditions for input 'g' - always set to zero

begin observer simulation

'q' and 'w' temporarily hold the current state estimate

pridict the measurement 'b_pri'. The pridiction is the current attitude rotation applied to the actual earth's magnetic field vector at that time.

calculate the measurement inovation 'v'

update the attitude estimate 'q' by rotating about the 'v' axis with a fixed gain.

renormalize the attitude estimate

estimate the next cycle's a priori state

generate output plot

Data Collection & IGRF Model (cont.)

Kalman Filter Routine (sat_ekf.m) - not working

=====

```
set the moment of inertia 'I' for later use by linear and
non-linear satellite models.

set sample rate in seconds 'T'

set damping 'd' - always set to zero

load earths magnetic field vector 'b_IGRF'

load magnetometer data 'b_mag' via the function 'getbmag'
    getbmag - reads and normalizes data file 'Uo22tel.txt'

set initial conditions for state estimate 'x'

set initial conditions for input 'g' - always set to zero

initialize the error covariance matrixes
    'Q' - Process error covariance
    'R' - Measurement error covariance
    'P' - Estimate error covariance

begin Kalman simulation

    'q' and 'w' temporarily hold the current state estimate

    pridict the measurement 'b_pri'. The pridiction is the
    current attitude rotation applied to the actual earth's
    magnetic field vector at that time.

    linearize about current state estimate and estimate the
    estimate error covariance 'P'

    calculate the Kalman filter gain 'K'

    update the state estimate covariance 'P'

    calculate the measurement inovation 'v'

    update the attitude estimate 'q' by rotating about the
    'v' axis.

    renormalize the attitude estimate

    update the rate estimate 'w' in traditional manner

    estimate the next cycle's a priori state

generate output plot
```

References

- [1] Lefferts, E. J., Markley, F. L., and Shuster, M. D., "Kalman Filtering for Spacecraft Attitude Estimation", *Journal of Guidance, Control, and Dynamics*, vol. 5, no. 5, pp.417-429, Sept.-Oct. 1982,.
- [2] Psiaki, M. L., Martel, F., and Pal, P. K., "Three-Axis Attitude Determination via Kalman Filtering of Magnetometer Data", *Journal of Guidance, Control, and Dynamics*, vol. 13, no. 3, pp.506-514, May-Jun. 1990.
- [3] Halliday, L. D., "Communications Infrastructure for the MOST Microsatellite Project", Master's thesis, University of Toronto, 2000.
- [4] Antsaklis, P. J. and Michel, A. N., *Linear Systems*, 1st ed. New York: McGraw-Hill, 1997, pp. 350-363.
- [5] O'Neil, P. V., *Advanced Engineering Mathematics*, 4th ed. Boston: PWS, 1995, pp. 208-210.
- [6] Wertz, J. R., *Spacecraft Attitude Determination and Control*, Holland: D. Reidel, 1978.
- [7] Emami, R.M., "Robot Kinematics and Dynamics", MIE 1062F Course Notes, University of Toronto, 1999.
- [8] Davidoff, M., *The Satellite Experimenter's Handbook*, 2nd ed. Connecticut: The American Radio Relay League, 1990.
- [9] Welch, G. and Bishop, G., "An Introduction to the Kalman Filter", TR 95-041, University of North Carolina, Department of Computer Science.
This paper can also be found at <http://www.cs.unc.edu/~welch/> and <http://www.cs.unc.edu/~gb>.
- [10] Wisniewski, R., "Satellite Attitude Control Using Only Electromagnetic Actuation", Ph.D. Thesis, Aalborg University, 1996.
This paper can also be found at <http://www.control.auc.dk/~raf/>.
- [11] Downs, L., "Using Quaternions to Represent Rotation", CS184 Course Notes, University of California, Berkeley, 2000.
This paper can also be found at <http://www.cs.berkeley.edu/~laura/cs184/quat/quaternion.html>.
- [12] Sweetser, D., "Doing Physics with Quaternions", independent dissertations, 1997.
This paper can also be found at <http://world.std.com/~sweetser/quaternions/qindex/qindex.html>.
- [13] Sakurai, A., "Attitude Determination for Coarse Pointing Operation", MOST Microsatellite Design Project, University of Toronto, 1999.
- [14] Magliacane, J. A., "Spotlight On: UoSAT-OSCAR-22", *The AMSAT Journal*, vol 15, no. 3, May/June 1992.
This paper can also be found at http://www.amsat.org/amsat/sats/n7hpr/uo22_kd2.html.
- [15] The Radio Amateur Satellite Corporation, "UoSAT-OSCAR 22 Satellite Summary", online document, January 31, 1999.
This document can be found at <http://www.amsat.org/amsat/sats/n7hpr/uo22.html>.
- [16] Surrey Space Centre, "Small Satellites Home Page", online document, January 29, 2000.
This document can be found at <http://www.ee.surrey.ac.uk/SSC/SSHP/micro91.html>.