Analysis of an Application where the Unscented Kalman Filter is Not Appropriate

Except where reference is made to the work of others, the work described in this dissertation is my own or was done in collaboration with my advisory committee. This dissertation does not include proprietary or classified information.

Abby Anderson

Certificate of Approval:

Robert Dean, Co-Chair
Assistant Professor
Electrical and Computer Engineering

A. Scottedward Hodel, Co-Chair
Associate Professor
Electrical and Computer Engineering

Stanley Reeves
Professor
Electrical and Computer Engineering

George Flowers
Professor
Mechanical Engineering

George Flowers
Dean
Graduate School

Analysis of an Application where the Unscented Kalman Filter is Not Appropriate

Abby Anderson

A Dissertation

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Doctor of Philosophy

Auburn, Alabama
December 18, 2009

Analysis of an Application where the Unscented Kalman Filter is Not Appropriate

Abby Anderson

Signature of Author

Date of Graduation

Dissertation Abstract

Analysis of an Application where the Unscented Kalman Filter is Not Appropriate

Abby Anderson

Doctor of Philosophy, December 18, 2009
(M.S., Auburn University, 2006)
(B.E.E., Auburn University, 2003)

241 Typed Pages

Directed by A. Scottedward Hodel and Robert Dean

In this work a spin stabilized rocket with a ring of lateral pulse jets for attitude correction and fins that open early in flight is simulated. The rocket is simulated with five different sensor packages: rate gyros only, rate gyros and an ideal magnetometer, rate gyros and a magnetometer, rate gyros and angle gyros, and rate gyros and angle gyros and a magnetometer. The gyros are microelectromechanical systems (MEMS) devices. All control effort must be applied in the first seconds of flight because of the properties of the rocket. A comparison of the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF) for each sensor suite is presented to determine the best approach to improve the circular probable error (CEP) of the rocket. A solution to Wahba's problem, the EStimator of the Optimal Quaternion (ESOQ) algorithm, is used for state estimation for certain sensor configurations. Wahba's problem has traditionally been used for state estimation of orbiting satellites.

Acknowledgments

First, I would like to thank my family. Particularly my parents Jeff and Rhonda Anderson. Without their support none of this would have been possible.

I would also like to thank my committee: Dr. Hodel, Dr. Dean, Dr. Reeves, and Dr. Flowers. I would especially like to thank Dr. Dean for stepping into a difficult situation to allow me to complete my work.

But the person who is most directly responsible for the completion of this Ph.D. is Dr. Hodel. (Yes, Dr. Hodel, I realize I started the previous sentence with a conjunction, but note that there are no dangling modifiers.) Over the years that I have been his student, we have had conversations ranging from work to our sometimes unorthodox religious views. No matter the topic, Babylon 5 always had a way of working itself into the discussion. As Delenn aptly said, "The universe puts us in places where we can learn. They are never easy places, but they are right. Wherever we are, it's the right place ... and the right time. Pain that sometimes comes is part of the process of constantly begin born." I miss our conversations. I have been blessed not only to have Dr. Hodel as an adviser for both my graduate degrees but also to consider him my friend.

One day in one of my undergraduate classes that Dr. Hodel was teaching, he said, "My students' job is to make me look good." I never quite let him forget that statement, but I hope this work (for which he suggested the title "The Unscented Kalman Filter Stinks") makes you look good, Dr. Hodel. It is dedicated to you.

"A student is not above his teacher, nor a servant above his master. It is enough for the student to be like his teacher and the servant like his master ..." (Matthew 10:24-25, NIV).

"And so it begins." - Kosh, Babylon 5

Style manual or journal used IEEE Transactions on Aerospace and Electronic Systems (together with the style known as "auphd"). Bibliography follows IEEE Transactions.

Computer software used The document preparation package TEX (specifically LATEX) together with the departmental style-file auphd.sty.

Table of Contents

List of Figures

Chapter 1
Introduction

Increasing the accuracy of munitions while reducing cost is of key interest to the military. One method is to use onboard guidance systems that can steer a missile to within meters or less of its target. However, this approach is expensive. A less costly but less accurate alternative is the spin stabilization approach. These systems lack closed-loop flight control and many of the sophisticated sensors found in onboard guidance systems. The accuracy of spin stabilized rockets is affected by many factors including motor misalignments, tip-off error, and wind.

One method to improve the accuracy of spin stabilized rockets is to include microelectromechanical system (MEMS) sensors. MEMS sensors have the advantages of low cost since they are batch producible using conventional IC technology and light weight. However, they are subject to various errors including constant biases, walking biases, and additive noise.

In this work we compare state estimation techniques to improve the accuracy of a spin stabilized rocket equipped with various sensor suites which include MEMS devices. We compare the improvement in rocket accuracy when states are estimated by a Kalman Filter (KF), an Extended Kalman Filter (EKF), and an Unscented Kalman Filter (UKF) for various sensor configurations. We also use the EStimator of the Optimal Quaternion (ESOQ) algorithm, a solution to Wahba's problem, alone and in conjunction with the EKF and UKF to estimate states with certain sensor suites. The first sensor configuration consists only of MEMS gyros to measure the rocket's rotational rates. The second sensor suite consists of MEMS gyros and an ideal sensor that measures a known inertial frame vector in the body frame. The third sensor suite is comprised of MEMS gyros and a tri-axial magnetometer. The next sensor suite consists of MEMS rate gyros and MEMS gyros that directly measure rotational angles. The final sensor suite combines the rate gyros, angle gyros, and magnetometer into one package.

For this work we make several assumptions about the rocket. The rocket is launched from a stationary ground launcher. A main thruster propels the rocket, and a ring of lateral thrusters located at the rocket's rear supply torque for control. Fins are located at the back of the rocket that open shortly after launch. Due to aerodynamic forces, control of such a rocket is only feasible during the first seconds of flight so state estimation is confined to that time interval.

In this dissertation we present previous work pertaining to improving munition accuracy, develop system models for the various sensor suites, and present the results of state estimation. In Chapter 2 we present several solutions to Wahba's problem, a method of determining a vehicle's attitude using pairs of sensors such as accelerometers and magnetometers. Next, in Chapter 3 we review concepts associated with navigation via magnetometers. In Chapter 4 we present methods for controlling rockets with a focus on rockets with lateral jets. We develop the system equations for our various rocket models in Chapter 5. We present the results of simulating the systems from Chapter 5 in Chapter 6 and draw some conclusions.

In this work we make the following contributions:

- We estimate the states of a rocket that is controlled solely with a ring of lateral thrusters.
- We concentrate on estimation of a rocket's states during the first seconds of flight. Other work focuses on controlling munitions later in flight.
- We compare the effects that various sensor suites have on munition accuracy.
- We use a magnetometer to estimate the states of a rocket. Magnetometers are conventionally used on satellites rather than rockets. Usually a satellite has a model of the magnetic field of the body it is orbiting on board. A rocket, however, does not have a magnetic field model available.
- We apply a solution of Wahba's problem to estimate the states of a rocket. Wahba's problem is traditionally applied to orbiting satellites since magnetic field and gravitational field measurements can be found with magnetometers and accelerometers.

- We compare the performances of various state estimators including the KF, EKF, UKF, and ESOQ algorithm to estimate the states of a rotating rocket and present conclusions about which estimator performs best.

Chapter 2
Wahba's Problem

Wahba's problem is a minimization problem most often applied to find the attitude of spacecraft with a fixed attitude. Measurements are taken in the spacecraft's body frame that give the angles between known objects and the spacecraft. Often the known objects are stars and the sensors are star trackers. The difference between the body frame measurements and known inertial frame values are used to derive an attitude solution for the spacecraft. The relationship between the body frame and the inertial frame is illustrated in Figure 2.1. The origin of the inertial frame (with axes denoted by a subscript i) is located at the center of the earth while the origin of the body frame (with axes denoted by a subscript b) is located at the center of gravity of the rocket. Various algorithms have been developed based on Wahba's cost function. While the cost function as originally posed is for static systems, researchers have extended it for use with dynamic systems.

2.1 Problem Statement

Grace Wahba first presented what has become known as Wahba's Problem in 1965 in *SIAM Review* [1]. The problem is as follows: Given two vector sets $\{\mathbf{v}_1,\ \mathbf{v}_2,\ \dots\ ,\mathbf{v}_n\}$ and $\{\mathbf{v}_1^*,\ \mathbf{v}_2^*,\ \dots\ ,\mathbf{v}_n^*\}$ with $n \geq 2$ entries, find the rotation matrix M (an orthogonal matrix with determinant $+1$) that minimizes

$$\sum_{j=1}^{n} \|\mathbf{v}_j^* - M\mathbf{v}_j\|^2.$$

In other words, find the rotation matrix M that minimizes the mean square error between the two vector sets. A solution to this problem means that given at least two noncolinear, nonzero measurements (such as magnetic field and gravitational acceleration) in the body frame of a moving vehicle and known values of the measured quantities in a stationary reference frame (i.e. an inertial reference frame), the attitude of the vehicle can be found via the rotation matrix M.

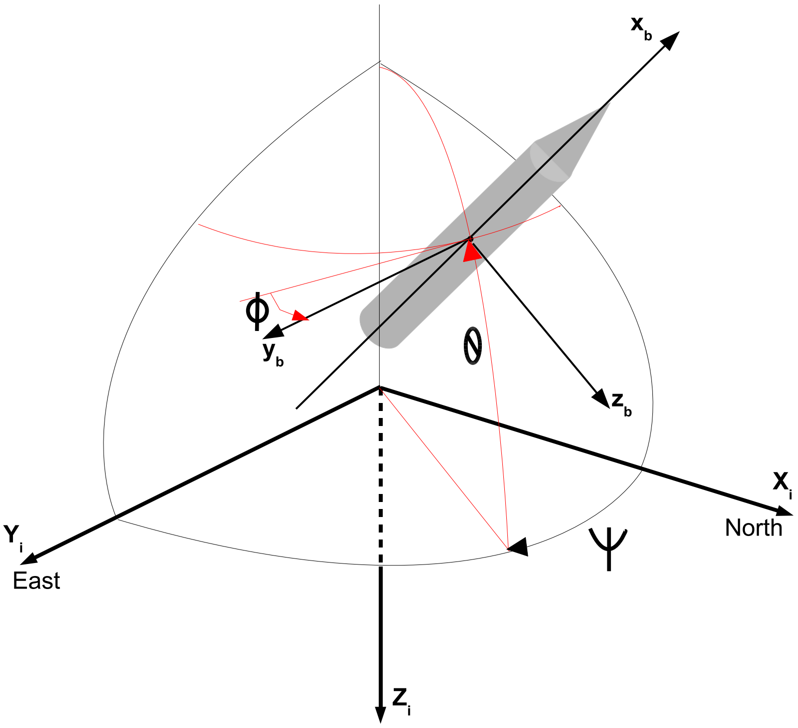

Figure 2.1: Inertial and Body Reference Frames

2.2 Problem Solutions

Various solutions to Wahba's Problem were published in 1966 [2]. The solution given by Farrell and Stuelpnagel, based on the polar decomposition, proceeds as follows. Let the column vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n, \mathbf{v}_1^*, \ldots, \mathbf{v}_n^*$ have dimension k, and let V and V^* denote the two $k \times n$ matrices formed from the two vector sets. Then the problem can be rewritten as

$$Q(M) = \sum_{j=1}^{n} \|\mathbf{v}_j^* - M\mathbf{v}_j\|^2 = \mathrm{tr}((V^* - MV)^T (V^* - MV)), \tag{2.1}$$

where $Q(M)$ is defined as the sum of squares to be minimized.

Proof. Let a $k \times n$ matrix A be defined as $[\mathbf{a}_1\ \mathbf{a}_2\ \ldots\ \mathbf{a}_n]$ where $\mathbf{a}_j$ are column vectors of length k. Then

$$A^T A = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_n^T \end{bmatrix} \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \ldots & \mathbf{a}_n \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1^T\mathbf{a}_1 & \mathbf{a}_1^T\mathbf{a}_2 & \ldots & \mathbf{a}_1^T\mathbf{a}_n \\ \mathbf{a}_2^T\mathbf{a}_1 & \mathbf{a}_2^T\mathbf{a}_2 & \ldots & \mathbf{a}_2^T\mathbf{a}_n \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{a}_n^T\mathbf{a}_1 & \mathbf{a}_n^T\mathbf{a}_2 & \ldots & \mathbf{a}_n^T\mathbf{a}_n \end{bmatrix}$$

$$\mathrm{tr}\{A^T A\} = \mathbf{a}_1^T\mathbf{a}_1 + \mathbf{a}_2^T\mathbf{a}_2 + \ldots + \mathbf{a}_n^T\mathbf{a}_n = \sum_{j=1}^{n} \|\mathbf{a}_j\|^2$$

□

Expand expression (2.1) so that $Q(M)$ becomes

$$\begin{aligned} Q(M) &= \mathrm{tr}((V^{*T} - V^T M^T)(V^* - MV)) \\ &= \mathrm{tr}(V^{*T}V^*) + \mathrm{tr}(V^{*T}MV) - \mathrm{tr}(V^T M^T V^*) + \mathrm{tr}(V^T M^T M V). \end{aligned}$$

Since M is orthogonal (i.e. $M^T M = MM^T = I$) and $\mathrm{tr}(ABC) = \mathrm{tr}(C^T B^T A^T)$, Q(M) can be written as

$$Q(M) = \mathrm{tr}(V^{*T}V^*) + \mathrm{tr}(V^T V) - 2\,\mathrm{tr}(V^T M^T V^*). \tag{2.2}$$

Since only the last term of Equation (2.2) is dependent on M, $Q(M)$ is minimized when $F(M) = \text{tr}(V^T M^T V^*)$ is maximized. $F(M)$ may be written as

$$F(M) = \text{tr}(M^T V^* V^T)$$

because $\text{tr}(ABC) = \text{tr}(BCA) = \text{tr}(CAB)$ when the matrices' dimensions are conformable. By the polar decomposition, $V^* V^T$ can be written as UP where U is orthogonal (and unique if $V^* V^T$ is nonsingular) and P is symmetric and positive semidefinite. So

$$F(M) = \text{tr}(M^T U P).$$

Because P is a real symmetric matrix, it has a spectral decomposition $P = NDN^T$ where N is an orthogonal matrix and $D = \text{diag}(d_1, \ldots, d_n)$ with $d_i \geq d_{i+1}$. Defining $X = NM^T UN^T$ results in

$$F(M) = \text{tr}(M^T UN^T DN) = \text{tr}(NM^T UN^T D) = \text{tr}(XD) = \sum_{i=1}^{k} d_i x_{ii}.$$

$F(M)$ attains its maximum value when the elements x_{ii} are all at their maximum value. Since X is an orthogonal matrix, all of its elements lie between the values 1 and -1. This means the diagonal elements are maximized when they are equal to 1, which means the $F(M)$ is the identity matrix.

Since M is a rotation matrix, its determinant is required to be 1. Thus the determinant of X is

$$|X| = |NM^T UN^T| = |N||M^T||U||N^T| = |N|^2|M||U| = |U|.$$

If $\det(U) = 1$, then $X = I$ maximizes $F(M)$. If $\det(U) = -1$, then $\det(X) = -1$, and a solution is

$$X = \begin{bmatrix} I_{k-1} & 0 \\ 0 & -1 \end{bmatrix}.$$

Define X_0 as the matrix which maximizes $F(M)$ according to the determinant of U so that $X_0 = NM_0^T UN^T$. Then the rotation matrix that minimizes the sum of squares of $Q(M)$ is

$$M_0 = UN^T X_0^T N.$$

The matrix is unique if $V^* V^T$ is nonsingular.

Wessner offers another solution to Wahba's problem [2]. Like Farrell and Stuelpnagel, Wessner recasts the problem to maximize

$$F(M) = \mathrm{tr}(M^T V^* V^T).$$

If $V^* V^T$ is nonsingular, then $V^* V^T$ has the polar decomposition

$$V^* V^T = A = UP.$$

Define $U = (A^T)^{-1}(A^T A)^{1/2}$ and $P = (A^T A)^{1/2}$ where $(A^T A)^{1/2}$ is the symmetric square root of $A^T A$ with positive eigenvalues. From Farrell and Stuelpnagel, the optimal solution is $M_o = UN^T X_o N$ where N is orthogonal. Wessner assumes that the determinant of A is positive; thus, $X_o = I$. Then $M_o = UN^T N = U$, which results in

$$\begin{aligned} M_o &= (A^T)^{-1}(A^T A)^{1/2} \\ &= (V^* V^T)^{-1}(V^* V^T V^* V^T)^{1/2} \end{aligned}$$

2.3 Davenport's q-Method

Another method for the solution of Wahba's problem is the q-method developed by Paul Davenport in 1968 [3]. Davenport defines a rotation vector to solve the problem. We discuss the q-method in two parts. We first present an overview of rotation vectors, followed by a discussion of Wahba's problem. Davenport often uses the notation X^2 to represent $X^T X$ where X is a vector. Thus, in Davenport's notation, X^2 is a scalar value not to be confused with the vector value X. For simplicity, we use Davenport's notation in this discussion.

2.3.1 Rotation Vectors

A rotation of an angle $0 \leq \theta \leq \pi$ about a unit vector $X = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \in \mathbb{R}^3$ can be represented as the 3×3 matrix operator

$$\begin{aligned} R_x(\theta) &= \cos(\theta) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + (1 - \cos(\theta)) \begin{bmatrix} x_1^2 & x_1x_2 & x_1x_3 \\ x_1x_2 & x_2^2 & x_2x_3 \\ x_1x_3 & x_2x_3 & x_3^2 \end{bmatrix} \\ &+ \sin(\theta) \begin{bmatrix} 0 & x_3 & -x_2 \\ -x_3 & 0 & x_1 \\ x_2 & -x_1 & 0 \end{bmatrix} \end{aligned} \qquad (2.3)$$

and

$$R = (B^T)^{-1} B \qquad (2.4)$$

where

$$B = \begin{bmatrix} 1 & x_3 \tan\left(\frac{\theta}{2}\right) & -x_2 \tan\left(\frac{\theta}{2}\right) \\ -x_3 \tan\left(\frac{\theta}{2}\right) & 1 & x_1 \tan\left(\frac{\theta}{2}\right) \\ x_2 \tan\left(\frac{\theta}{2}\right) & -x_1 \tan\left(\frac{\theta}{2}\right) & 1 \end{bmatrix}. \qquad (2.5)$$

In order to simplify notation, define

$$Y = \tan\left(\frac{\theta}{2}\right) X \qquad \text{and} \qquad Z = \sin\left(\frac{\theta}{2}\right) X$$

from which we obtain $\sin\left(\frac{\theta}{2}\right) = \frac{\sqrt{Y^2}}{\sqrt{1+Y^2}} = \sqrt{Z^2}$, $\cos\left(\frac{\theta}{2}\right) = \frac{1}{\sqrt{1+Y^2}} = \sqrt{1 - Z^2}$, $\sin(\theta) = \frac{2\sqrt{Y^2}}{1+Y^2} = 2\sqrt{Z^2(1 - Z^2)}$, and $\cos(\theta) = \frac{1-Y^2}{1+Y^2} = 1 - 2Z^2$. Hence Equation (2.3) can be rewritten as

$$R = \frac{1}{1+Y^2} \left((1 - Y^2) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + 2 \begin{bmatrix} y_1^2 & y_1y_2 & y_1y_3 \\ y_1y_2 & y_2^2 & y_2y_3 \\ y_1y_3 & y_2y_3 & y_2^2 \end{bmatrix} + 2 \begin{bmatrix} 0 & y_3 & -y_2 \\ -y_3 & 0 & y_1 \\ y_2 & -y_1 & 0 \end{bmatrix} \right) \qquad (2.6)$$

or

$$R = (1 - 2Z^2)\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + 2\begin{bmatrix} z_1^2 & z_1z_2 & z_1z_3 \\ z_1z_2 & z_2^2 & z_2z_3 \\ z_1z_3 & z_2z_3 & z_3^2 \end{bmatrix} + 2\sqrt{1-Z^2}\begin{bmatrix} 0 & z_3 & -z_2 \\ -z_3 & 0 & z_1 \\ z_2 & -z_1 & 0 \end{bmatrix}. \quad (2.7)$$

Alternatively, B may be written in terms of the elements of the vector Y as

$$B = \begin{bmatrix} 1 & y_3 & -y_2 \\ -y_3 & 1 & y_1 \\ y_2 & -y_1 & 1 \end{bmatrix}. \quad (2.8)$$

Let R be a rotation matrix, and separate R into symmetric and skew symmetric parts

$$R = \frac{1}{2}(R + R^T) + \frac{1}{2}(R - R^T) \quad (2.9)$$

where $R + R^T$ is symmetric and $R - R^T$ is skew symmetric. Then a comparison of Equation (2.9) to Equation (2.7) yields nine conditions for the vector Z. Taking the trace σ of Equation (2.7) yields

$$\sigma = 3 - 4Z^2 = 1 + 2\cos(\theta) \quad (2.10)$$

and inspection of the skew symmetric portion of R yields

$$2\sqrt{1-Z^2}Z = \frac{1}{2}\begin{bmatrix} r_{23} - r_{32} \\ r_{31} - r_{13} \\ r_{12} - r_{21} \end{bmatrix}. \quad (2.11)$$

Rearrange Equations (2.10) and (2.11) to obtain

$$Z^2 = \frac{3-\sigma}{4} \qquad \text{and} \qquad Z = \frac{1}{2\sqrt{1+\sigma}}\begin{bmatrix} r_{23} - r_{32} \\ r_{31} - r_{13} \\ r_{12} - r_{21} \end{bmatrix}.$$

We write Z^2 in terms of θ which results in

$$Z^2 = \frac{1}{2} - \frac{1}{2}\cos(\theta),$$

which means that $Z^2 \leq 1$. Thus, the mapping defined by Equation (2.7) is a mapping of three-dimensional vectors over the field of real numbers whose Euclidean length is less than or equal to one (denote this set of vectors by ζ) onto the group of rotation matrices. The mapping is one-to-one except when $Z^2 = 1$ ($\sigma = -1$).

Since $Y = \frac{Z}{\cos\left(\frac{\theta}{2}\right)}$, $Y = \frac{1}{\sqrt{1-Z^2}}Z$. Then using the same methods as above

$$Y = \frac{1}{1+\sigma}\begin{bmatrix} r_{23} - r_{32} \\ r_{31} - r_{13} \\ r_{12} - r_{21} \end{bmatrix}.$$

Y is undefined when $\sigma = -1$. To avoid this singularity, allow vectors of infinite magnitude whose direction is given by a unit vector X. When $\sigma = -1$, Equation (2.6) reduces to

$$R = -I + 2XX^T. \tag{2.12}$$

Let η denote the set of all real three-dimensional vectors augmented by the vectors of infinite magnitude just discussed. Then Equations (2.6) and (2.12) define a mapping from η onto the group of rotation matrices. Thus, either ζ or η may be used to parametrize the group of rotations. To distinguish between the new type of vector and ordinary vectors, Davenport defines the rotation vector: Given a set δ of real three-dimensional vectors and a mapping τ that maps δ onto the group of rotation matrices, then two elements of δ are said to be equivalent if they map into the same rotation matrix. Normal vector operations are applied to rotation vectors.

Rotation vectors can be combined to yield new rotation vectors. Given two rotation vectors Y_1 and Y_2, there are two associated rotation matrices R_1 and R_2. Then $R = R_2R_1$ is also a rotation matrix, and there exists an associated rotation vector Y. This vector Y is given by

$$Y = \frac{1}{1 - Y_1 Y_2} \left(Y_1 + Y_2 + Y_1 \times Y_2 \right). \tag{2.13}$$

Similarly, for the two rotation vectors Z_1 and Z_2 which define rotations R_1 and R_2, there exists a Z that gives $R = R_2 R_2$. Z is defined by

$$\begin{aligned} Z &= \operatorname{sgn}\left(\sqrt{1 - Z_1^2}\sqrt{1 - Z_2^2} - Z_1 \cdot Z_2 \right) Z_0 && (2.14a) \\ Z_0 &= \sqrt{1 - Z_2^2} Z_1 + \sqrt{1 - Z_1^2} Z_2 + Z_1 \times Z_2. && (2.14b) \end{aligned}$$

This leads to the definition of Davenport's rotation product: Let δ be a set of vectors, let $*$ be a binary operation on δ, and let τ be a mapping of δ onto the group of rotation matrices. Then $*$ is said to be a rotation product if it is preserved by τ, i.e. if $\tau(V * W) = \tau(V)\tau(W)$ for every V and W in δ.

Davenport also develops other useful relationships between rotation matrices, vectors, and rotation vectors. Assume a vector V is rotated by R to yield $V' = RV$. If Y and Z are the rotation vectors defining R, then V' is

$$\begin{aligned} V' &= YV \\ &= \frac{1}{1 + Y^2}[(1 - Y^2)V + 2(V \cdot Y)Y + 2V \times Y], \quad Y^2 < \infty && (2.15a) \\ &= -V + 2(V \cdot X)X, \quad Y^2 = \infty && (2.15b) \end{aligned}$$

$$V' = ZV = (1 - 2Z^2)V + 2(V \cdot Z)Z + 2\sqrt{1 - Z^2} V \times Z. \tag{2.16}$$

To determine R from V' and V where $V' = RV$

$$\begin{aligned} Y &= \frac{1}{1 + V \cdot V'} V' \times V, \quad V \cdot V' \neq -1 && (2.17) \\ Z &= \frac{1}{\sqrt{2(1 + V \cdot V')}} V' \times V, \quad V \cdot V' \neq -1 && (2.18) \end{aligned}$$

If $V \cdot V' = -1$, then Z is any vector satisfying the two conditions $Z^2 = 1$ and $Z \cdot V = 0$ and Y has infinite magnitude with direction defined by Z.

2.3.2 q-Method

Davenport poses Wahba's problem in the following manner

$$\begin{aligned}\phi(R) &= \sum_{i=1}^{n}(W_i - RV_i)^2 \\ &= \sum_{i=1}^{n}[W_i^2 + V_i^2 - 2W_i \cdot (RV_i)]\end{aligned}$$

where V_i is a vector in the inertial frame and W_i is a vector in the body frame. Recasting the problem in terms of the rotation vector Y results in

$$\begin{aligned}\phi(Y) &= \sum_{i=1}^{n}(V_i^2 + W_i^2 - \frac{2}{1+Y^2}[(1-Y^2)V_i \cdot W_i + 2(V_i \cdot Y)(W_i \cdot Y) \\ &+ 2(W_i \times V_i) \cdot Y]), \quad Y^2 < \infty \qquad (2.19a) \\ &= \sum_{i=1}^{n}\left(V_i^2 + W_i^2 - 2[-V_i \cdot W_i + 2(V_i \cdot X)(W_i \cdot X)]\right), \quad Y^2 = \infty \qquad (2.19b)\end{aligned}$$

where $X^2 = 1$. Take the derivative of Equation (2.19a) with respect to y_j, where j refers to the j^{th} component of a vector, to obtain

$$\begin{aligned}\frac{\partial \phi}{\partial y_j} = \frac{-4}{(1+Y^2)^2}\sum_{i=1}^{n}(2[(V_i \times W_i) \cdot Y - (V_i \cdot Y)(W_i \cdot Y) - V_i W_i]y_j \\ + (1+Y^2)[(V_i \cdot Y)w_{ij} + (W_i \cdot Y)v_{ij} - (V_i \times W_i)_j]).\end{aligned}$$

Thus, for ϕ to have a minimum, we must have

$$2\sum_{i=1}^{n}[(W_i \times V_i) \cdot Y + (V_i \cdot Y)(W_i \cdot Y) + V_i \cdot W_i]Y = (1+Y^2)\sum_{i=0}^{n}[(V_i \cdot Y)W_i + (W_i \cdot Y)V_i + W_i \times V_i]. \quad (2.20)$$

Take the dot product with Y of both sides and rearrange terms, which results in

$$2\sum_{i=1}^{n}(V_i \cdot Y)(W_i \cdot Y) = \sum_{i=1}^{n}[(W_i \times V_i) \cdot Y + 2V_i \cdot W_i]Y - \sum_{i=1}^{n}(W_i \times V_i) \cdot Y \tag{2.21}$$

Substituting Equation (2.21) into Equation (2.19a) results in the expression

$$\phi(R) = \sum_{i=1}^{n}(V_i^2 + W_i^2 - 2V_i \cdot W_i) - 2Y \cdot \sum_{i=1}^{n} W_i \times V_i$$

which implies that ϕ is minimized when $2Y \cdot \sum_{i=1}^{n} W_i \times V_i$ is maximized. Substituting Equation (2.21) into Equation (2.20) yields

$$\sum_{i=1}^{n}\left([(W_i \times V_i) \cdot Y + 2V_i \cdot W_i]Y - [(V_i \cdot Y)W_i + (W_i \cdot Y)V_i + W_i \times V_i]\right) = 0$$

which can be written as

$$(A^T Y I + B)Y = A \tag{2.22}$$

where I is identity, A is the vector $\sum_{i=1}^{n} W_i \times V_i$, and B is a symmetric matrix with elements

$$b_{jk} = -\sum_{i=1}^{n}(v_{ij}w_{ik} + v_{ik}w_{ij}), \quad j \neq k \tag{2.23}$$

$$b_{jj} = 2\sum_{i=1}^{n}(V_i \cdot W_i - v_{ij}w_{ij}). \tag{2.24}$$

If $A = 0$, then $Y = 0$ is the desired solution. For $A \neq 0$, multiply each side of Equation (2.22) by the adjoint of $(A^T Y I + B)$ to yield

$$\det(A^T Y I + B)Y = \operatorname{adj}(A^T Y I + B)A.$$

Then multiply the result by A^T to get

$$\det(A^T Y I + B)A^T Y = A^T \operatorname{adj}(A^T Y I + B)A.$$

Define the scalar $A^T = \lambda$, $\det(\lambda I + B) = f(\lambda)$, and $A^T \operatorname{adj}(\lambda I + B)A = g(\lambda)$. Then

$$h(\lambda) = \lambda f(\lambda) - g(\lambda) = 0.$$

The maximum value of $Y \cdot \sum_{i=1}^{n}(W_i \times V_i)$ (which minimizes $\phi(Y)$) is the largest zero of $h(\lambda)$, λ_0, where $h(\lambda)$ is a fourth-degree polynomial in λ and $-f(\lambda)$ is the characteristic polynomial of B. Since $\phi(Y)$ is a nonnegative function,

$$\lambda_0 \leq \frac{1}{2}\sum_{i=1}^{n}(V_i - W_i)^2,$$

which gives an upper bound on λ_0.

Since B is symmetric, an orthogonal matrix P exists such that $P^{-1}BP$ is diagonal. Let Y' and U be vectors such that $Y = PY'$ and $A = PU$. Then

$$((U^T Y')I + B)PY' = PU.$$

Multiply both sides by P^{-1} to get

$$((U^T Y')I + D)Y' = U$$

where D is a diagonal matrix with the eigenvalues of B arranged in increasing order ($\lambda_1 \leq \lambda_2 \leq \lambda_3$) as its entries. Multiply by $\operatorname{adj}((U^T Y')I + D)$ and then by U^T to get

$$\det(U^T Y' I + D)U^T Y' = U^T \operatorname{adj}(U^T Y' I + D)U.$$

Since $U^T Y' = A^T Y = \lambda$ and D is diagonal,

$$\lambda(\lambda + \lambda_1)(\lambda + \lambda_2)(\lambda + \lambda_3) = u_1^2(\lambda + \lambda_2)(\lambda + \lambda_3) + u_2^2(\lambda + \lambda_1)(\lambda + \lambda_3) + u_3^2(\lambda + \lambda_1)(\lambda + \lambda_2).$$

Some analysis on this equation yields a lower bound for λ_0:

$$\lambda_0 \geq -\lambda_1$$

A numerical algorithm can then be used to find λ_0.

Since the above analysis is derived from Equation (2.19a), the solution is only valid for rotations of less than 180°. To obtain the minimum among all 180° rotations, a Lagrange multiplier term is added to the derivative of Equation (2.19b) with respect to x_j, which yields

$$-4\sum_{i=1}^{n}(V_i \cdot X)w_{ij} + (W_i \cdot X)v_{ij} + 2\mu x_j = 0, \quad j = 1, 2, 3.$$

Writing the above equation in terms of a single vector equation results in

$$\left(\left(\frac{\mu}{2} - 2\sum_{i=1}^{n} V_i \cdot W_i\right) I + B\right) X = 0 \tag{2.25}$$

where the matrix B is as defined in Equations (2.23)-(2.24). The roots of Equation (2.25) are

$$\frac{\mu_k}{2} - 2\sum_{i=1}^{n} V_i \cdot W_i = \lambda_k, \quad k = 1, 2, 3$$

where the λ_k's are the eigenvalues of B. The condition equations then become

$$(B - \lambda_k I)X = 0, \quad X^2 = 1,$$

and the solutions are the unit eigenvectors of B. Then the solution $\phi(R)$ for rotations of 180° is

$$\phi(R) = \sum_{i=1}^{n}(V_i - W_i)^2 + 2\lambda_k$$

and

$$\phi(R) = \sum_{i=1}^{n}(V_i - W_i)^2 - 2\lambda_0$$

for rotations other than 180°. Thus, the rotation minimizing $\phi(R)$ is

$$Y = (\lambda_0 I + B)^{-1} A$$

when $f(\lambda_0) \neq 0$ or

$$(\lambda_0 I + B)X = 0 \quad \text{and} \quad X^2 = 1$$

when $f(\lambda_0) = 0$, where λ_0 is the largest zero of $h(\lambda)$.

For the case where $f(\lambda_0)$ is near zero, a near-singular matrix occurs. This can be avoided by using the Z vector representation. This is given by

$$Z = \frac{\operatorname{sgn}(f(\lambda_0))}{\sqrt{f^2(\lambda_0) + A^T(\operatorname{adj}(\lambda_0 I + B))^2 A}} \operatorname{adj}(\lambda_0 I + B)A$$

when $f(\lambda_0) \neq 0$ and

$$(\lambda_0 I + B)Z = 0 \quad \text{and} \quad Z^2 = 1$$

when $f(\lambda_0) = 0$ where

$$\begin{aligned}
\operatorname{adj}(\lambda I + B) &= \lambda^2 I + \lambda(\operatorname{tr}(B)I - B) + \operatorname{adj}(B) \\
f(\lambda) &= \lambda^3 + \operatorname{tr}(B)\lambda^2 + \operatorname{tr}(\operatorname{adj}(B))\lambda + \det(B) \\
g(\lambda) &= A^T \operatorname{adj}(\lambda I + B)A.
\end{aligned}$$

For the case where $n = 2$, $V_1^2 = V_2^2$, and $W_1^2 = W_2^2$, the least squares solution is found as follows. Define

$$\begin{aligned}
U_1 &= \frac{V_1 - V_2}{|V_1 - V_2|} \\
U_1' &= \frac{W_1 - W_2}{|W_1 - W_2|} \\
U_2 &= \frac{V_1 + V_2}{|V_1 + V_2|} \\
U_2' &= \frac{W_1 + W_2}{|W_1 + W_2|},
\end{aligned}$$

and use the rotation vector techniques to obtain the rotation that maps U_i to U_i' to yield the least squares solution.

A concise summary of Davenport's q-method is given in [4]. Wahba's problem is presented as

$$J(A) = \sum_{i=1}^{n} w_i \|\hat{u}_B^i - A\hat{u}_R^i\|^2$$

where w_i is the weight of the ith vector measurement, $\hat{u}_B^i$ are inertial frame vectors, $\hat{u}_R^i$ are the corresponding vectors in the body frame, and A is a rotation matrix. The cost function may also be written as

$$J(A) = -2\sum_{i=1}^{n} W_i A V_i + \text{constant terms}$$

where the unnormalized vectors W_i and V_i are defined as

$$W_i = \sqrt{w_i}\hat{u}_B^i \quad V_i = \sqrt{w_i}\hat{u}_R^i.$$

The loss function is a minimum when

$$J'(A) = \sum_{i=1}^{n} W_i A V_i = \text{tr}(W^T A V)$$
$$W = \begin{bmatrix} W_1 & W_2 & \dots & W_n \end{bmatrix}$$
$$V = \begin{bmatrix} V_1 & V_2 & \dots & V_n \end{bmatrix}$$

and is maximized.

To find the attitude matrix A, write A in terms of the quaternion q

$$A(q) = (q_4^2 - \mathbf{q} \cdot \mathbf{q})I + 2\mathbf{q}\mathbf{q}^T - 2q_4 Q$$

where

$$q = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix}$$

$$Q = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}.$$

Substitute $A(q)$ into the cost function to yield

$$J'(q) = q^T K q \tag{2.26}$$

where

$$\begin{aligned} K &= \begin{bmatrix} S - I\sigma & Z \\ Z^T & \sigma \end{bmatrix} \\ B &= WV^T \\ S &= B^T + B \\ Z &= \begin{bmatrix} B_{23} - B_{32} & B_{31} - B_{13} & B_{12} - B_{21} \end{bmatrix}^T \\ \sigma &= \mathrm{tr}(B). \end{aligned}$$

The extrema of J', subject to the constraint $q^T q = 1$, can be found using Lagrange multipliers. Define

$$g(q) = q^T K q - \lambda q^T q$$

where λ is the Lagrange multiplier. Differentiating the equation with respect to q^T and setting the result equal to zero results in

$$Kq = \lambda q. \tag{2.27}$$

Thus, the quaternion that gives the optimal attitude is an eigenvector of K. Substitute Equation (2.27) into Equation (2.26) to get

$$J'(q) = q^T K q = q^T \lambda q = \lambda.$$

Thus, J' is maximized when the eigenvector corresponding to the largest eigenvalue is chosen.

2.4 QUEST Algorithm

The quaternion estimation (QUEST) algorithm is a method to solve Wahba's problem. The algorithm is a single-point algorithm so attitude estimates are based solely on current measurements. The following is a summary of the QUEST algorithm as presented in [5].

Wahba's problem can be posed in terms of quaternions as

$$J(q) = \frac{1}{2} \sum_{i=1}^{k} a_i \|\mathbf{b}_i - A(q)\mathbf{r}_i\|^2 \tag{2.28}$$

where $\mathbf{b}_i$ are measured body frame unit vectors, $\mathbf{r}_i$ are the corresponding vectors in the inertial frame, and a_i are positive weights. The goal is to find the quaternion q that minimizes the cost function J. The minimization of Equation (2.28) can be written as the maximization of the cost function $g(q)$

$$g(q) = 1 - J(q) / \left(\frac{1}{2} \sum_{i=1}^{k} a_i \right),$$

which can be written as

$$g(q) = q^T K q. \tag{2.29}$$

The matrix K is formed as follows. Define

$$\begin{aligned} m_k &= \sum_{i}^{k} a_i && (2.30a) \\ \sigma &= \frac{1}{m_k} \sum_{i=1}^{k} a_i \mathbf{b}_i^T \mathbf{r}_i && (2.30b) \end{aligned}$$

$$\begin{aligned} B &= \frac{1}{m_k}\sum_{i=1}^{k} a_i \mathbf{b}_i \mathbf{r}_i^T && (2.30c) \\ S &= B + B^T && (2.30d) \end{aligned}$$

$$\begin{aligned} \mathbf{z} &= \frac{1}{m_k}\sum_{i=1}^{k} a_i (\mathbf{b}_i \times \mathbf{r}_i) && (2.30e) \\ K &= \begin{bmatrix} S - \sigma I & \mathbf{z} \\ \mathbf{z}^T & \sigma \end{bmatrix}. && (2.30f) \end{aligned}$$

The optimal unit quaternion q^* that maximizes $g(q)$ satisfies

$$Kq^* = \lambda q^*$$

where λ is a Lagrange multiplier. Then λ is an eigenvalue of K and q^* is an eigenvector. Thus, $g(q^*) = \lambda$. Because it is desired to maximize g, choose $\lambda = \lambda_{MAX}$, the largest eigenvalue of K, and q^* as the corresponding eigenvector. Rodrigues parameters (or Gibbs vector) can be used to calculate the eigenvector:

$$\begin{aligned} \mathbf{y}^* &= [(\lambda_{MAX} + \sigma)I - S]^{-1}\mathbf{z} && (2.31a) \\ q^* &= \frac{1}{\sqrt{1 + |\mathbf{y}^*|^2}} \begin{bmatrix} \mathbf{y}^* \\ 1 \end{bmatrix}. && (2.31b) \end{aligned}$$

The maximum eigenvalue is exactly one when error free measurements are used; otherwise the eigenvalue is close to one.

2.5 Kalman Filter Type Approach

In [6] and [7] Shuster develops the basis to cast the QUEST algorithm as a Kalman filter-type problem. In [6] it is shown that Wahba's problem can be equivalent to maximum likelihood estimation through the appropriate choice of weights. In [7] Shuster extends the results of [6] to form a filter QUEST algorithm that is comparable to a Kalman filter. [6] is summarized in §2.5.1 and [7] in §2.5.2.

2.5.1 Wahba's Problem as Maximum Likelihood Estimation

Wahba's loss function can be written as

$$L(A) = \frac{1}{2}\sum_{k=1}^{n} a_k \|\hat{W}_k - A\hat{V}_k\|^2$$

where $\hat{W}_k$ are body-fixed unit vectors and $\hat{V}_k$ are the same unit vectors in an inertial frame. Assume $\hat{W}_k$ is the measurement provided by sensor k and take the unit vector measurement to have the probability density

$$\rho_{\hat{W}_k}(\hat{W}_k', A) = \mathcal{N}_k \exp\left(-\frac{1}{2\sigma_k^2}|\hat{W}_k' - A\hat{V}_k|^2\right) \tag{2.32}$$

which is defined over the unit sphere

$$|\hat{W}_k'| = 1. \tag{2.33}$$

$\hat{W}_k'$ is the value of the random variable $\hat{W}_k$ which satisfies Equation (2.33), and $\mathcal{N}_k$ is chosen to ensure that the probability density function is valid:

$$\begin{aligned} \mathcal{N}_k^{-1} &= \int_0^{2\pi}\int_0^{\pi} \exp\left(-\frac{1}{2\sigma_k^2}(1-\cos(\theta))\right)\sin(\theta)d\theta d\phi \\ \mathcal{N}_k &= [2\pi\sigma_k^2(1-e^{-2/\sigma_k^2})]^{-1}. \end{aligned} \tag{2.34}$$

The probability density function is about the direction $A\hat{V}_k$ and can be approximated by a tangent plane

$$W_k = A\hat{V}_k + \Delta W_k, \quad \Delta W_k \cdot A\hat{V}_k = 0.$$

The sensor error $\Delta\hat{W}_k$ is approximately Gaussian with

$$E[\Delta W_k] = 0, \quad E[\Delta W_k \Delta W_k^T] = \sigma_k^2[I - (A\hat{V}_k)(A\hat{V}_k)^T]$$

It can be shown that for $\hat{W}_k$

$$\begin{aligned} E[\Delta\hat{W}_k] &= -\rho_k^2\tau_k A\hat{V}_k \\ E[\Delta\hat{W}_k\Delta\hat{W}_k] &= \rho_k^2[I-(3-2\tau_k)(A\hat{V}_k)(A\hat{V}_k)^T] \\ \rho_k^2 &= \frac{1}{2}E[|\hat{W}_k\times(A\hat{V}_k)|^2] \\ \tau_k &= \frac{1}{\rho_k^2}E[1-\hat{W}_k\cdot A\hat{V}_k]. \end{aligned}$$

which means that

$$\begin{aligned} \rho_k^2 &= \sigma_k^2-\sigma_k^4+O(e^{-2/\sigma_k^2}) \\ \tau_k &= \frac{1}{1-\sigma_k^2}+O(e^{-2/\sigma_k^2}). \end{aligned}$$

Define Z'_k as a sequence of measurements and $\rho_{z_1,\ldots,z_n}(Z'_1,\ldots Z'_n,x)$ as a joint probability distribution where x is a parameter vector. Then for the model given by Equation (2.32)

$$\begin{aligned} \rho_{z_1,\ldots,z_n}(Z'_1,\ldots,Z'_n,A) &= \prod_{k=1}^{n}\frac{1}{2\pi\sigma_k^2 f_k}\exp(-\|\hat{W}_k-A\hat{V}_k\|^2/2\sigma_k^2) \\ f_k &= 1-e^{-2/\sigma_k^2}. \end{aligned}$$

Define the negative-log-likelihood function as

$$J(x) = -\log\left(\rho_{z_1,\ldots,z_n}(Z'_1,\ldots,Z'_n,x)\right)$$

so that

$$J(A)=\sum_{k=1}^{n}\left(\frac{1}{2\sigma_k^2}\|\hat{W}'_k-A\hat{V}_k\|^2+\log\sigma_k^2+\log 2\pi+\log f_k\right). \tag{2.35}$$

Allowing $a_k=\frac{1}{\sigma_k^2}$, the negative-log-likelihood function becomes equivalent to the Wahba loss function.

The QUEST algorithm gives an estimate of attitude. To get an estimate of the error covariance matrix, the Fisher information matrix is used. The Fisher information matrix for x is defined by

$$F_{xx} = E\left[\frac{\partial^2}{\partial x \partial x^T} J(x)\right]_{x_{true}}.$$

As more data is gathered, the Fisher information matrix approaches the inverse of the error covariance matrix

$$\lim_{n\to\infty} F_{xx} = P_{xx}^{-1}.$$

The Fisher information matrix is not defined in terms of the quaternion, but rather in terms of incremental error angles θ for which

$$A = e^{[[\theta]]} A_{true}$$

where the notation $[[\mathbf{x}]]$ denotes the matrix

$$[[\mathbf{x}]] = \begin{bmatrix} 0 & x_3 & -x_2 \\ -x_3 & 0 & -x_1 \\ x_2 & -x_1 & 0 \end{bmatrix}$$

for some vector $\mathbf{x}$. Then

$$J(\theta) = \lambda_{MAX}^{(0)} - \mathrm{tr}(e^{[[\theta]]} A_{true} B^T)$$

which leads to

$$\begin{aligned} F_{\theta\theta} &= \mathrm{tr}(A_{true} B_{true}^T) I - A_{true} B_{true}^T \\ &= \sum_{k=1}^{n} \frac{1}{\sigma_k^2} (I - (\hat{W}_k)_{true} (\hat{W}_k^T)_{true}) \end{aligned}$$

where

$$(\hat{W}_k)_{true} = A_{true} \hat{V}_k.$$

Evaluated at the maximum likelihood estimate of the attitude, $F_{\theta\theta}$ is approximated as

$$F_{\theta\theta} \approx \operatorname{tr}(A^{*'}_{ML} B^{'T})I - A^{*'}_{ML} B^{'T}.$$

Then

$$B' = \left[\frac{1}{2}\operatorname{tr}(F_{\theta\theta})I - F_{\theta\theta}\right] A^{*'}_{ML}.$$

2.5.2 Filter QUEST Algorithm

The attitude estimate A^*_k based on the first k estimates and the error covariance matrix $P_{k|k}$ are determined by

$$B_k = \sum_{l=1}^{k} a_l \hat{W}_l \hat{V}_l^T.$$

For the reasons presented in §2.5.1 Shuster chooses $a_l = 1/\sigma_l^2$. Thus,

$$B_k = B_{k-1} + \frac{1}{\sigma_k^2}\hat{W}_k \hat{V}_k^T. \tag{2.36}$$

When A_k is constant, Equation (2.36) is the filter.

For dynamic systems

$$\frac{d}{dt} = [[\omega(t)]]A(t).$$

Let $A(t_k)$ satisfy $A_{k+1} = \phi_k A_k$ with known ϕ_k. Define

$$\frac{d}{dt}\hat{W}_k(t) = [[\omega(t)]]\hat{W}_k(t)$$

with boundary condition $\hat{W}_k(t) = \hat{W}_k$ where $\hat{W}_k$ is the time t_k measurement. So

$$\hat{W}_l(t_{k+1}) = \phi_k \hat{W}_l(t_k).$$

Then B_k at time t_k is

$$B_k = \sum_{l=1}^{k} \frac{1}{\sigma_l^2}\phi_{k-1}\phi_{k-2}\ldots\phi_l \hat{W}_l \hat{V}_l^T$$

which satisfies

$$B_k = \phi_{k-1} B_{k-1} + \frac{1}{\sigma_k^2} \hat{W}_k \hat{V}_k^T.$$

Define

$$\begin{aligned} B_{k|k-1} &= B(t_k) \quad \text{given} \quad \hat{W}_1, \ldots, \hat{W}_{k-1} \\ &= \sum_{l=1}^{k-1} \frac{1}{\sigma_l^2} \hat{W}_l(t_k) \hat{V}_l^T \\ B_{k|k} &= B(t_k) \quad \text{given} \quad \hat{W}_1, \ldots, \hat{W}_k \\ &= \sum_{l=1}^{k} \frac{1}{\sigma_l^2} \hat{W}_l(t_k) \hat{V}_l^T. \end{aligned}$$

Then the filter QUEST algorithm is

$$\begin{aligned} B_{k|k} &= \phi_{k-1} B_{k-1|k-1} && (2.37a) \\ &= B_{k|k-1} + \frac{1}{\sigma_k^2} \hat{W}_k \hat{V}_k^T. && (2.37b) \end{aligned}$$

Information from the previous attitude $A^*_{0|0}$ and corresponding estimate error covariance matrix $P_{0|0}$ can be included

$$B_{0|0} = \left[\frac{1}{2} \operatorname{tr}(P_{0|0}^{-1}) I - P_{0|0}^{-1} \right] A^*_{0|0}. \tag{2.38}$$

Otherwise, if no *a priori* information is available

$$B_{0|0} = 0. \tag{2.39}$$

Direct incorporation of process noise into the QUEST algorithm problem statement results in prohibitively expensive computational costs in the resulting algorithm. An alternative approach is to use a fading memory approximation. With this method the filter QUEST formulation becomes

$$\begin{aligned} B_{0|0} &= \left[\frac{1}{2}\operatorname{tr}(P_{0|0}^{-1})I - P_{0|0}^{-1}\right] A_{0|0}^{*} \quad \text{or} \quad 0 && (2.40) \\ B_{k+1|k} &= \alpha\phi_k B_{k|k} && (2.41) \\ B_{k|k} &= B_{k|k-1} + \frac{1}{\sigma_k^2}\hat{W}_k\hat{V}_k^T. && (2.42) \end{aligned}$$

For $\alpha = 1$ the usual QUEST algorithm results and for $\alpha = 0$ only the current measurements contribute to the estimate. Notice that unlike a traditional Kalman filter, the QUEST algorithm computes only attitude. It does not include estimates of biases or misalignments.

2.6 REQUEST Algorithm

The REQUEST (recursive quaternion estimation) algorithm, developed by Bar-Itzhack [5], is a recursive version of the QUEST algorithm. Rather than updating the attitude profile matrix B as Shuster does in the filter QUEST algorithm [7], the REQUEST algorithm recursively updates the QUEST algorithm's K matrix. An algorithm is developed for both the time-invariant case and time-varying case.

2.6.1 Time-Invariant REQUEST

For the time-invariant case, the vehicle's body frame axes are assumed to be nonrotating with respect to the reference axes. The algorithm begins by first processing k sets of vectors with the QUEST algorithm. Let K_k denote the K matrix computed by the QUEST algorithm. It is desired to update K with j new pairs of vectors. Then define $\mathbf{b}_i$ as measured body frame unit vectors, $\mathbf{r}_i$

as the corresponding vectors in the inertial frame, and a_i as positive weights, and let

$$\begin{aligned}
\delta m_{k+j} &= \sum_{i=k+1}^{k+j} a_i && (2.43a)\\
\delta \sigma_{k+j} &= \sum_{i=k+1}^{k+j} a_i \mathbf{b}_i^T \mathbf{r}_i && (2.43b)\\
\delta B_{k+j} &= \sum_{i=k+1}^{k+j} a_i \mathbf{b}_i \mathbf{r}_i^T && (2.43c)\\
\delta S_{k+j} &= \delta B_{k+j} + \delta B_{k+j}^T && (2.43d)\\
\delta \mathbf{z}_{k+j} &= \sum_{i=k+1}^{k+j} a_i (\mathbf{b}_i \times \mathbf{r}_i) && (2.43e)
\end{aligned}$$

$$\delta K_{k+j} = \begin{bmatrix} \delta S_{k+j} - \delta\sigma_{k+j} I & \delta \mathbf{z}_{k+j} \\ \delta \mathbf{z}_{k+j}^T & \delta \sigma_{k+j} \end{bmatrix}. \qquad (2.43f)$$

Then the new algorithm is

$$\begin{aligned}
m_{k+j} &= m_k + \delta m_{k+j} && (2.44a)\\
K_{k+j} &= (m_k/m_{k+j}) K_k + (1/m_{k+j}) \delta K_{k+j}. && (2.44b)
\end{aligned}$$

The maximum eigenvalue of K_{k+j} is then calculated, and Equations (2.31a)-(2.31b) are used to find the optimal attitude quaternion.

2.6.2 Time-Varying REQUEST

In the time-varying case, the body frame has rotation with respect to the reference frame. Bar-Itzhack develops the REQUEST algorithm [5] both for the case of error-free measurements (i.e. no process noise) and for the case of noisy measurements.

Error-Free Propagation

Assume that at time t_n, k pairs of vectors $\mathbf{b}_i$ and $\mathbf{r}_i$ are processed using the QUEST algorithm. Immediately afterward, the body frame axes rotate to a new orientation and at time t_{n+1} j new vector measurements are taken. It is desired to find the attitude quaternion estimate $q^*_{n+1|n}$ at time t_{n+1} based on the first k measurements. Then the cost function $g(q) = q^T K q$ can be written as

$$g(q_{n|n}) = q^T_{n|n} K_{n|n} q_{n|n}.$$

The dynamics of a rotation are described by the differential equation

$$\dot{q} = \frac{1}{2}\Omega q \tag{2.45}$$

where Ω is a skew symmetric matrix of angular velocities

$$\Omega = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix}.$$

Then

$$q(t_{n+1}) = \Phi(t_{n+1}, t_n) q(t_n)$$

where $\Phi(t_{n+1}, t_n)$ is the state transition matrix. Set $q(t_n) = q_{n|n}$, where $q_{n|n}$ is the quaternion to be transformed from time t_n to time t_{n+1}, and $q(t_{n+1}) = q_{n+1|n}$ where $q_{n+1|n}$ is the transformed quaternion $q_{n|n}$ to get

$$q_{n+1|n} = \Phi q_{n|n}.$$

Because Ω is skew symmetric, Φ is orthogonal so

$$q_{n|n} = \Phi^T q_{n+1|n}.$$

Then

$$g(q_{n|n}) = g'(q_{n+1|n}) = q_{n+1|n}^T \Phi K_{n|n} \Phi^T q_{n+1|n}.$$

Thus, the problem has been transformed into finding the quaternion estimate $q_{n+1|n}$ that maximizes g'. Let

$$K_{n+1|n} = \Phi K_{n|n} \Phi^T.$$

Since $K_{n+1|n}$ is obtained through a similarity transform, it has the same eigenvalues as $K_{n|n}$. Then

$$g'(q_{n+1|n}) = q_{n+1|n}^T K_{n+1|n} q_{n+1|n}.$$

Adding the constraint that $|q_{n+1|n}|^2 = 1$ by using the Lagrange multiplier $\lambda_{n+1|n}$, $q_{n+1|n}^*$ satisfies the equation

$$K_{n+1|n} q_{n+1|n}^* = \lambda_{n+1|n} q_{n+1|n}^*$$

and is the eigenvector of $K_{n|n}$ that corresponds to the largest eigenvalue. So the algorithm becomes

$$K_{n+1|n} = \Phi K_{n|n} \Phi^T \tag{2.46a}$$

$$K_{n+1|n+1} = (m_n/m_{n+1}) K_{n+1|n} + (1/m_{n+1}) \delta K_{n+1} \tag{2.46b}$$

where $m_n = m_k$ and m_{n+1} and δK_{n+1} are calculated using Equations (2.44a) and (2.43f), respectively.

Noisy Propagation

For the case of noiseless propagation, it is assumed that the rotation rates ω are known perfectly. That is, there is no process noise, only noise from the sensors. With noisy propagation, the measured angular rate can be described as

$$\omega_m = \omega + \epsilon$$

where ϵ is the error component. Bar-Itzhack assumes that the gyros are of good quality with slowly varying biases. To handle noisy measurements, two different possible modifications to the

REQUEST algorithm are proposed:

$$K_{n+1|n+1} = \rho_n(m_n/m_{n+1})(K_{n+1|n} + 1/m_{n+1})\delta K_{n+1}$$

and

$$K_{n+1|n+1} = \frac{\rho_n m_n}{p_n m_n + \delta m_{n+1}} K_{n+1|n} + \frac{1}{\rho_n m_n + \delta m_{n+1}} \delta K_{n+1}$$

where $0 < \rho_n \leq 1$ is a forgetting factor to determine how much weight is placed on past measurements.

2.7 Extended QUEST Algorithm

The QUEST algorithm and the previously presented algorithms based on QUEST have the disadvantage of being unable to estimate anything other than attitude. Other dynamic states - for example, gyro bias - cannot be estimated. In [8] Psiaki presents an extension of the QUEST algorithm called Extended QUEST that attempts to overcome these limitations.

Yet another way to write Wahba's loss function as given by Equation (2.28) is

$$J_{QUEST}[A(q)] = \sum_{i=1}^{m} \frac{1}{\sigma_i^2} + \frac{1}{2} q^T H_{meas} q \tag{2.47a}$$

where the symmetric Hessian matrix H_{meas} is given by

$$H_{meas} = \sum_{i=1}^{m} \frac{2}{\sigma_i^2} \begin{bmatrix} [I(\mathbf{b}_i^T \mathbf{r}_i) - \mathbf{r}_i \mathbf{b}_i^T - \mathbf{b}_i \mathbf{r}_i^T] & -(\mathbf{b}_i \times \mathbf{r}_i) \\ -(\mathbf{b}_i \times \mathbf{r}_i)^T & -\mathbf{b}_i^T \mathbf{r}_i \end{bmatrix} \tag{2.47b}$$

and $\mathbf{r}_i$ is a set of known unit vectors in the inertial frame, $\mathbf{b}_i$ is the set $\mathbf{r}_i$ measured in the body frame, q is the quaternion that rotates from the inertial frame to the body frame, and σ_i are weights. Differentiating Equation (2.47a) with respect to q and adding the constraint $q^T q = 1$ with a Lagrange multiplier λ results in

$$(H_{meas} + \lambda I)q = 0 \quad \text{or} \quad H_{meas} q = -\lambda q.$$

q is both a quaternion and a normalized eigenvector of H_{meas}, and $-\lambda$ is the corresponding eigenvalue. The optimal solution to the problem occurs when q corresponds to the most negative $-\lambda$.

The extended QUEST filter solves a more general quadratic function of the form

$$\begin{aligned} \min_q J(q) &= \frac{1}{2} q^T H q + g^T q && (2.48a) \\ \text{subject to} \qquad & q^T q = 1 && (2.48b) \end{aligned}$$

where H is the cost function's Hessian matrix and g is the cost function's gradient vector at $q = 0$. Solving the problem results in

$$\begin{aligned} J'(q) &= \frac{1}{2} q^T H q + g^T q + \frac{1}{2} \lambda q^T q \\ \frac{\partial J'(q)}{\partial q} &= 0 = Hq + g + \lambda I q \\ 0 &= (H + \lambda I) q + g. \end{aligned}$$

Solving for q results in $q = -(H + \lambda I)^{-1} g$, which can be substituted into Equation (2.48b) to yield

$$g^T (H + \lambda I)^{-2} g = 1. \qquad (2.49)$$

Multiplying both sides of Equation (2.49) by $(\det(H + \lambda I))^2$ results in an eighth-order polynomial in λ from which the optimal (largest) λ can be found using numerical algorithms.

The problem statement for the extended QUEST algorithm is then

$$\begin{aligned} \min_{q(k), \mathbf{x}(k)} J &= \frac{1}{2} \sum_{i=1}^{m(k)} \frac{1}{\sigma_i^2(k)} \{\mathbf{b}_i(k) - A[q(k)]\mathbf{r}_i(k)\}^T \{\mathbf{b}_i(k) - A[q(k)]\mathbf{r}_i(k)\} \\ &+ \frac{1}{2} [R_{ww}(k-1)\mathbf{w}(k-1)]^T [R_{ww}(k-1)\mathbf{w}(k-1)] \\ &+ \frac{1}{2} \{R_{qq}(k-1)[q(k-1) - \hat{q}(k-1)]\}^T \{R_{qq}(k-1)[q(k-1) - \hat{q}(k-1)]\} \\ &+ \frac{1}{2} \{R_{xq}(k-1)[q(k-1) - \hat{q}(k-1)] + R_{xx}(k-1)[\mathbf{x}(k-1) - \hat{\mathbf{x}}(k-1)]\}^T \\ &\times \{R_{xq}(k-1)[q(k-1) - \hat{q}(k-1)] \\ &+ R_{xx}(k-1)[\mathbf{x}(k-1) - \hat{\mathbf{x}}(k-1)]\} && (2.50a) \end{aligned}$$

subject to

$$\begin{aligned}
q(k) &= \Phi[t(k), t(k-1); q(k-1), \mathbf{x}(k-1), \mathbf{w}(k-1)]q(k-1) && (2.50b) \\
x(k) &= f_x[t(k), t(k-1); q(k-1), \mathbf{x}(k-1), \mathbf{w}(k-1)] && (2.50c) \\
q^T(k)q(k) &= 1 && (2.50d)
\end{aligned}$$

where q is the attitude quaternion, $\mathbf{x}$ is the vector of filter states, and $\mathbf{w}$ is the process noise vector. The vectors $\hat{q}(k-1)$ and $\hat{\mathbf{x}}(k-1)$ are the *a posteriori* estimates of q and $\mathbf{x}$ at sample time $t(k-1)$. The matrices $R_{ww}(k-1)$, $R_{qq}(k-1)$, $R_{xq}(k-1)$, and $R_{xx}(k-1)$ are penalizing weights. Equations (2.50b) and (2.50c) are the filter's dynamic model. Φ is the state transition matrix associated with the quaternion's kinematic differential equation given by Equation (2.45).

The first of the two parts of the extended QUEST algorithm is the propagation step. This step begins with obtaining *a priori* estimates of $q(k)$ and $\mathbf{x}(k)$

$$\begin{aligned}
\tilde{q}(k) &= \Phi[t(k), t(k-1); \hat{q}(k-1), \hat{\mathbf{x}}(k-1), 0]\hat{q}(k-1) && (2.51) \\
\tilde{\mathbf{x}}(k) &= f_x[t(k), t(k-1); \hat{q}(k-1), \hat{\mathbf{x}}(k-1), 0]. && (2.52)
\end{aligned}$$

The next step in part one is to develop a linearized dynamic model from Equations (2.51)-(2.52)

$$\begin{bmatrix} \Delta q(k) \\ \Delta\mathbf{x}(k) \end{bmatrix} = \begin{bmatrix} \Phi_{qq}(k-1) & \Phi_{qx}(k-1) \\ \Phi_{xq}(k-1) & \Phi_{xx}(k-1) \end{bmatrix} \begin{bmatrix} \Delta q(k-1) \\ \Delta\mathbf{x}(k-1) \end{bmatrix} + \begin{bmatrix} \Gamma_q(k-1) \\ \Gamma_x(k-1) \end{bmatrix} w(k-1)$$

where

$$\Delta\mathbf{q}(k) = q(k) - \tilde{q}(k)$$

$$\Delta\mathbf{x}(k) = \mathbf{x}(k) - \tilde{\mathbf{x}}(k)$$

$$\Delta\mathbf{q}(k-1) = q(k-1) - \hat{q}(k-1)$$

$$\Delta\mathbf{x}(k-1) = \mathbf{x}(k-1) - \hat{\mathbf{x}}(k-1)$$

$$
\begin{aligned}
\Phi_{qq}(k-1) &= \Phi + \left[\frac{\partial \Phi}{\partial q}\right] \hat{q}(k-1) \\
\Phi_{qx}(k-1) &= \left[\frac{\partial \Phi}{\partial x}\right] \hat{q}(k-1) \\
\Gamma_{q}(k-1) &= \left[\frac{\partial \Phi}{\partial w}\right] \hat{q}(k-1)
\end{aligned}
$$

$$
\begin{aligned}
\Phi_{xq}(k-1) &= \frac{\partial f_x}{\partial q} \\
\Phi_{xx} &= \frac{\partial f_x}{\partial x} \\
\Gamma_{x}(k-1) &= \frac{\partial f_x}{\partial w}
\end{aligned}
$$

with all partial fractions evaluated at $[q(k-1), \mathbf{x}(k-1), \mathbf{w}(k-1)] = [\hat{q}(k-1), \hat{\mathbf{x}}(k-1), 0]$. $\boldsymbol{\Delta}\mathbf{q}(k-1)$ and $\boldsymbol{\Delta}\mathbf{q}(k)$ are not quaternions. The final propagation step is to form an information matrix and left QR-factorize

$$
Q \begin{bmatrix} \tilde{R}_{qq}(k) & 0 & 0 \\ \tilde{R}_{xq}(k) & \tilde{R}_{xx}(k) & 0 \\ \tilde{R}_{wq}(k) & \tilde{R}_{wk}(k) & \tilde{R}_{ww}(k-1) \end{bmatrix} = \begin{bmatrix} R_1 & R_2 & R_3 \\ 0 & 0 & R_{ww}(k-1) \end{bmatrix}
$$

where

$$
\begin{aligned}
R_1 &= \begin{bmatrix} R_{qq}(k-1) & 0 \\ R_{xq}(k-1) & R_{xx}(k-1) \end{bmatrix} \\
R_2 &= \begin{bmatrix} \Phi_{qq}(k-1) & \Phi_{qx}(k-1) \\ \Phi_{xq}(k-1) & \Phi_{xx}(k-1) \end{bmatrix}^{-1} \\
R_3 &= \begin{bmatrix} I & 0 & -\Gamma_q(k-1) \\ 0 & I & -\Gamma_x(k-1) \end{bmatrix}.
\end{aligned}
$$

The matrices $\tilde{R}_{qq}(k)$, $\tilde{R}_{xx}(k)$, and $\tilde{R}_{ww}(k{-}1)$ are square, and $\tilde{R}_{xx}(k)$ and $\tilde{R}_{ww}(k{-}1)$ are nonsingular. The propagation step results in the modified cost function

$$\begin{aligned} J &= \frac{1}{2}\sum_{i=1}^{m(k)} \frac{1}{\sigma^2_{i(k)}}\{\mathbf{b}_i(k) - A[q(k)]\mathbf{r}_i(k)\}^T\{\mathbf{b}_i(k) - A[q(k)]\mathbf{r}_i(k)\} \\ &+ \frac{1}{2}\{\tilde{R}_{qq}(k)[q(k) - \tilde{q}(k)]\}^T\{\tilde{R}_{qq}(k)[q(k) - \tilde{q}(k)]\} \\ &\cdot \frac{1}{2}\{\tilde{R}_{xq}(k)[q(k) - \tilde{q}(k)] + \tilde{R}_{xx}(k)[\mathbf{x}(k) - \tilde{\mathbf{x}}(k)]\}^T \\ &\times \{\tilde{R}_{xq}(k)[q(k) - \tilde{q}(k)] + \tilde{R}_{xx}(k)[\mathbf{x}(k) - \tilde{\mathbf{x}}(k)]\}. \end{aligned} \tag{2.53}$$

It is assumed that $\mathbf{w}(k-1)$ is set to

$$[\mathbf{w}(k-1)]_{opt} = -\tilde{R}^{-1}_{ww}(k-1)\{\tilde{R}_{wq}(k)[q(k) - \tilde{q}(k)] + \tilde{R}_{wx}(k)[\mathbf{x}(k) - \tilde{\mathbf{x}}(k)]\}.$$

The second phase of the algorithm is a measurement update. Equations $(2.47a)$ and $(2.47b)$ express the squared measurement error cost terms quadratically in $q(k)$. The resulting measurement error Hessian matrix $H_{meas}(k)$ is used to pose the measurement update problem as

$$\begin{aligned} \min_{q(k),\mathbf{x}(k)} J &= \frac{1}{2}q^T(k)H_{meas}(k)q(k) + \frac{1}{2}\{\tilde{R}_{qq}(k)[q(k) - \tilde{q}(k)]\}^T\{\tilde{R}_{qq}(k)[q(k) - \tilde{q}(k)]\} \\ &\cdot \frac{1}{2}\{\tilde{R}_{xq}(k)[q(k) - \tilde{q}(k)] + \tilde{R}_{xx}(kk)[\mathbf{x}(k) - \tilde{\mathbf{x}}(k)]\}^T \\ &\times \{\tilde{R}_{xq}(k)[q(k) - \tilde{q}(k)] + \tilde{R}_{xx}(k)[\mathbf{x}(k) - \tilde{\mathbf{x}}(k)]\} \end{aligned} \tag{2.54a}$$

subject to

$$q^T(k)q(k) = 1. \tag{2.54b}$$

To find the optimum filter state estimate $\mathbf{x}(k)$, set the derivative of J with respect to x to zero to get

$$[\mathbf{x}(k)]_{opt} = \tilde{\mathbf{x}}(k) - \tilde{R}^{-1}_{xx}(k)\tilde{R}_{xq}[q(k) - \tilde{q}(k)]. \tag{2.55}$$

A least squares optimization problem results from substituting Equation (2.55) into the cost function

$$\begin{aligned} \min_{q(k)} J &= \frac{1}{2} q^T(k) H_{meas}(k) q(k) + \frac{1}{2}\{\tilde{R}_{qq}[q(k) - \tilde{q}(k)]\}^T \\ &\times \ \{\tilde{R}_{qq}(k)[q(k) - \tilde{q}(k)]\} \qquad (2.56a) \\ \text{subject to} & \\ q^T(k) q(k) &= 1, \qquad (2.56b) \end{aligned}$$

which is equivalent to Equation (2.47a) as seen by letting

$$H = H_{meas} + \tilde{R}_{qq}^T(k)\tilde{R}_{qq}(k) \qquad (2.57)$$

and

$$g = -\tilde{R}_{qq}^T(k)\tilde{R}_{qq}(k)\tilde{q}(k). \qquad (2.58)$$

The measurement update is then complete by solving Equation (2.47a) with Equations (2.57) and (2.58). The solution is the *a posteriori* estimate of the attitude quaternion at sample k, $\hat{q}(k)$. $\hat{q}(k)$ is substituted into Equation (2.55) to compute $\hat{\mathbf{x}}(k)$.

To apply the algorithm recursively, it is necessary to express the cost function as

$$\begin{aligned} J = &\frac{1}{2}\{R_{qq}(k)[q(k) - \hat{q}(k)]\}^T\{R_{qq}(k)[q(k) - \hat{q}(k)]\} \\ &+ \frac{1}{2}\{R_{xq}(k)[q(k) - \hat{q}(k)] + R_{xx}[\mathbf{x}(k) - \hat{\mathbf{x}}(k)]\}^T \\ &\times \{R_{xq}(k)[q(k) - \hat{q}(k)] + R_{xx}[\mathbf{x}(k) - \hat{\mathbf{x}}(k)]\} \end{aligned}$$

where $R_{qq}(k)$ is a matrix square root $R_{qq}^T(k)R_{qq}(k) = [H_{meas}(k) + \tilde{R}_{qq}^T(k)\tilde{R}_{qq}(k) + \lambda(k)I]$, $R_{xq}(k) = \tilde{R}_{xq}(k)$, and $R_{xx}(k) = \tilde{R}_{xx}(k)$. The matrix square root exists since $[H_{meas}(k) + \tilde{R}_{qq}^T(k)\tilde{R}_{qq}(k) + \lambda(k)I]$ is at least positive semidefinite at the global minimum.

2.8 Energy Approach Algorithm

In [9] Mortari develops several solutions to Wahba's problem through a unique physical interpretation. Mortari poses Wahba's problem as

$$\sigma_w = \frac{1}{2}\sum_{i=1}^{n} \alpha_i \|\mathbf{s}_i - A\mathbf{v}_i\|^2 = 1 - \sum_i \alpha_i \mathbf{s}_i^T A\mathbf{v}_i \tag{2.59}$$

where $\sum_i \alpha_i = 1$ and $\mathbf{s}_i$ and $\mathbf{v}_i$ are body frame and inertial frame unit vectors, respectively. The weights α_i are derived from a measure of the sensors' accuracy. Let

$$\beta_i = \frac{1}{\mathrm{E}[\cos^{-1}(S_i^T \mathbf{s}_i)]}$$

where S_i is a unit vector of true values and $\mathbf{s}_i$ is a unit vector of observed values as defined previously. Then $\alpha_i = \beta_i / \sum_k \beta_k$. It is assumed that errors in observed directions are less than 0.5° so that the angles ϑ_i between $\mathbf{s}_i$ and $A\mathbf{v}_i$ (where $\mathbf{s}_i^T A\mathbf{v}_i = \cos(\vartheta_i)$) can be considered small ($\cos(\vartheta_i) \approx 1 - \vartheta_i^2/2$). Then Equation (2.59) can be expressed as

$$\begin{aligned} \sigma_w &= 1 - \sum_i \alpha_i \mathbf{s}_i^T A\mathbf{v}_i = 1 - \sum_i \alpha_i \cos(\vartheta_i) \\ &\approx 1 - \sum_i \alpha_i (1 - \vartheta_i^2/2) = \frac{1}{2}\sum_i \alpha_i \vartheta_i^2. \end{aligned} \tag{2.60}$$

The expression in Equation (2.60) is the same as the expression for the energy in n springs with displacements ϑ_i and stiffnesses α_i. Applying the rotation matrix A to the $\mathbf{v}_i$ unit vectors is like a rigid rotation with the rotation being optimal when the rotated unit vectors are as close as possible to the sensed vectors as if they were attracted with a torque proportional to α_i. The more accurate the sensors are (the greater α_i), the more strongly $A\mathbf{v}_i$ is attracted. Therefore, the $\mathbf{v}_i$ can be thought of as a rigid body free to rotate about the origin and constrained by n spherical springs with stiffnesses α_i applied between the directions $A\mathbf{v}_i$ and $\mathbf{s}_i$. This formulation is identical to Wahba's problem.

The minimum value of stored spring energy $\mathcal{E}$ is defined from Equation (2.60)

$$\mathcal{E} = \frac{1}{2}\sum_i \alpha_i \vartheta_i^2.$$

Then

$$\mathcal{E} \approx \sigma_w = 1 - \sum_i \alpha_i s_i^T A v_i = 1 - \sigma_M.$$

$\mathcal{E}$ is minimized when σ_M is maximized

$$\begin{aligned} \sigma_M &= \sum_i \alpha_i s_i^T A v_i = \operatorname{tr}(AB^T) \\ B &= \sum_i \alpha_i s_i v_i^T. \end{aligned}$$

Since A is a rotation matrix, the cost function must include the constraint $A^T A = I$ where $\det(A) = 1$. The augmented cost function is then

$$\sigma_M^* = \operatorname{tr}(AB^T) - \operatorname{tr}\left(\frac{1}{2}L(A^T A - I)\right) = \operatorname{tr}\left(AB^T - \frac{1}{2}L(A^T A - I)\right)$$

where L is a symmetric matrix of Lagrange multipliers. The maximization of σ_M^* leads to one of the following expressions (depending on the method used):

$$L = \begin{cases} B^T A \\ BA^T. \end{cases} \tag{2.61}$$

Since L is symmetric, $B^T A = A^T B$ and $BA^T = AB^T$. Since $A^T = A^{-1}$, both expressions of L lead to

$$AB^T A = B, \tag{2.62}$$

which Mortari calls the R-equation. All solutions of the R-equation, A, are orthonormal.

One method to solve the R-equation is an eigenanalysis of the 6×6 matrix

$$H = \begin{bmatrix} 0 & B \\ B^T & 0 \end{bmatrix}.$$

This is just one formulation of the H-matrix; other formulations give alternative solutions. Since it is symmetric, H has only real eigenvalues. Also, if λ_i is an eigenvalue of H, then so is $-\lambda_i$. Let $\mathbf{u}_i$ be the upper three elements of the i^{th} eigenvector of H and $\mathbf{d}_i$ be the lower three elements. Then λ_i has eigenvector $\begin{bmatrix} \mathbf{u}_i \\ \mathbf{d}_i \end{bmatrix}$ and $-\lambda_i$ has eigenvector $\begin{bmatrix} -\mathbf{u}_i \\ \mathbf{d}_i \end{bmatrix}$. It is assumed that $0 \le \lambda_1 \le \lambda_2 \le \lambda_3$. Let $U = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{bmatrix}$, $D = \begin{bmatrix} \mathbf{d}_1 & \mathbf{d}_2 & \mathbf{d}_3 \end{bmatrix}$, and $\Lambda = \text{diag}(\lambda_1, \lambda_2, \lambda_3)$. Then

$$\begin{bmatrix} 0 & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} U \\ D \end{bmatrix} = \begin{bmatrix} U \\ D \end{bmatrix} \Lambda$$

$$BD = U\Lambda$$

$$B^T U = D\Lambda$$

$$U^{-1}BD = \Lambda.$$

So

$$B^T U = DU^{-1}BD \Rightarrow (UD)^{-1}B^T(UD^{-1}) = B.$$

Thus, $A = UD^{-1}$ satisfies the R-equation.

It can be shown that $2U^TU = I$ and $2D^TD = I$ where multiplying by 2 ensures orthonormality. Since H is symmetric, its eigenvectors $\mathbf{w}_i = \{\mathbf{u}_i^T \mathbf{d}_i^T\}$ are orthonormal. So

$$\mathbf{w}_i^T \mathbf{w}_k = \mathbf{u}_i^T \mathbf{u}_k + \mathbf{d}_i^T \mathbf{d}_k = \delta_{ik} \quad \delta_{ik} = \begin{cases} 1, & i = k \\ 0, & i \ne k. \end{cases}$$

When $i = k$, $\mathbf{u}_i^T\mathbf{u}_i + \mathbf{d}_i^T\mathbf{d}_i = 1$ for λ_i, and $-\mathbf{u}_i^T\mathbf{u}_i + \mathbf{d}_i^T\mathbf{d}_i = 0$ for $-\lambda_i$. Then if $\left[\begin{array}{cc} u_i^T & d_i^T \end{array}\right]^T$ is an eigenvector, then $\left[\begin{array}{cc} u_i^T & -d_i^T \end{array}\right]^T$ is an eigenvector as well. Further, these two eigenvectors are orthogonal to each other. So $\mathbf{d}_i^T\mathbf{d}_i = \mathbf{u}_i^T\mathbf{u}_i$, which means $2\mathbf{u}_i^T\mathbf{u}_i = 1$ and $2\mathbf{d}_i^T\mathbf{d}_i = 1$. So

$$\mathbf{u}_i^T\mathbf{u}_i = \mathbf{d}_i^T\mathbf{d}_i = \frac{1}{2}.$$

The matrices U and D are orthogonal. Also

$$B^T(BD) = B^T(U\Lambda) = D\Lambda^2$$
$$B(B^TU) = B(D\Lambda) = U\Lambda^2,$$

which means

$$(B^TB)D \quad = \quad D\Lambda^2 \qquad (2.63a)$$
$$(BB^T)U \quad = \quad U\Lambda^2. \qquad (2.63b)$$

This implies that

$$2U^TU = 2D^TD = I. \qquad (2.64)$$

So

$$A_I = 2UD^T. \qquad (2.65)$$

Note that Equations (2.63a)-(2.63b) represent the singular value decomposition of the B matrix ($B = U\Lambda D^T$).

The solution form given by Equation (2.65) has a singularity. It occurs if $\det(B) = 0$, which always happens when $n = 2$ and in some cases when $n > 2$. When $\det(B) = 0$, $\lambda_1 = 0$, and the eigenvectors associated with the eigenvalues $\pm\lambda_1 = 0$ cannot be discriminated. This can be overcome due to the orthogonality of U and D. Once $(\mathbf{u}_2, \mathbf{d}_2)$ and $(\mathbf{u}_3, \mathbf{d}_3)$ are found

$$\mathbf{u}_1 = \sqrt{2}\mathbf{u}_2 \times \mathbf{u}_3 \quad \mathbf{d}_1 = \sqrt{2}\mathbf{d}_2 \times \mathbf{d}_3.$$

An alternate solution form that does not have singularities can be found by performing an eigenanalysis on Equations (2.63a)-(2.63b). The resulting U and D matrices are orthonormal, and the conditions of Equation (2.64) do not hold. The first alternative solution form is

$$A_{II} = UD^{-1} = UD^T.$$

The eigenvectors $U = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{bmatrix}$ and $D = \begin{bmatrix} \mathbf{d}_1 & \mathbf{d}_2 & \mathbf{d}_3 \end{bmatrix}$ must refer to the same eigenvalues sequence $0 \leq \lambda_1 \leq \lambda_2 \leq \lambda_3$, and the condition $\det(U) = \det(D)$ must be satisfied.

Since BB^T and B^TB have the same eigenvalues, let $M = BB^T$ (or $M = B^TB$). Then the characteristic equation of M is

$$\lambda^3 + c_2\lambda^2 + c_1\lambda + c_0$$

where

$$\begin{aligned}
c_2 &= -\operatorname{tr}(M) = -m_{11} - m_{22} - m_{33}\\
c_1 &= \operatorname{tr}(\operatorname{adj}(M)) = m_{11}m_{22} + m_{11}m_{33} + m_{22}m_{33} - m_{12}^2 - m_{13}^2 - m_{23}^2\\
c_0 &= -\det(M) = m_{11}m_{22}m_{13} + 2m_{12}m_{13}m_{23} - m_{22}^2m_{13}^2 - m_{11}m_{23}^2 - m_{33}m_{12}^2.
\end{aligned}$$

M is symmetric so all of its eigenvalues are real. Setting

$$\begin{aligned}
p &= \sqrt{(c_2/3)^2 - c_1/3}\\
q &= [c_1/2 - (c_2/3)^2]c_2/3 - c_0/2\\
z &= \frac{1}{3}\cos^{-1}(q/p^3),
\end{aligned}$$

then

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = -p \begin{bmatrix} \cos(z) + \sqrt{3}\sin(z) \\ \cos(z) - \sqrt{3}\sin(z) \\ -2\cos(z) \end{bmatrix} - \frac{c_2}{3}.$$

When $n = 2$, $c_0 = -\det(M) = 0$. Then $(M - \lambda_i I)\mathbf{f}_i = 0$ where $\mathbf{f}_i$ is $\mathbf{u}_i$ or $\mathbf{d}_i$ depending on the choice of M. The row vectors of $(M - \lambda_i I)$ must lie on the same plane. The solution $\mathbf{f}_i$ can be computed by normalizing the cross product between two row vectors of the matrix $(M - \lambda_i I)$.

It is desired to compute $\mathbf{f}_i$ from the cross product having the highest modulus. The 3×3 symmetric matrix $(M - \lambda_i I)$ can be written as

$$M - \lambda_i I = \begin{bmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \mathbf{m}_3^T \end{bmatrix} = \begin{bmatrix} m_a & m_x & m_y \\ m_x & m_b & m_z \\ m_y & m_z & m_c \end{bmatrix}.$$

Then the eigenvector f_i is chosen from among

$$\begin{aligned} \mathbf{e}_1 = \mathbf{m}_2 \times \mathbf{m}_1 &= \begin{bmatrix} m_b m_c - m_z^2 & m_y m_z - m_x m_c & m_x m_z - m_y m_b \end{bmatrix}^T \\ \mathbf{e}_2 = \mathbf{m}_3 \times \mathbf{m}_1 &= \begin{bmatrix} m_y m_z - m_x m_c & m_a m_c - m_y^3 & m_x m_y - m_z m_a \end{bmatrix}^T \\ \mathbf{e}_3 = \mathbf{m}_1 \times \mathbf{m}_2 &= \begin{bmatrix} m_x m_z - m_y m_b & m_x m_y - m_z m_a & m_a m_b - m_x^2 \end{bmatrix}^T \end{aligned}$$

which are parallel. The $\mathbf{e}_i$ with highest modulus has the maximum element p_k

$$\begin{aligned} p_1 &= (m_b m_c - m_z^2)^2 \\ p_2 &= (m_a m_c - m_y^2)^2 \\ p_3 &= (m_a m_b - m_x^2)^2. \end{aligned}$$

Then

$$\mathbf{f}_i = \mathbf{e}_k / \sqrt{\mathbf{e}_k^T \mathbf{e}_k}.$$

By this method the evaluation of the eigenvectors $\mathbf{f}_2$ and $\mathbf{f}_3$ is sufficient. Then $\mathbf{f}_1 = \mathbf{f}_2 \times \mathbf{f}_3$.

Mortari gives several other possible solution forms. Other previously known solutions can be derived from the R-equation as well. Appropriate manipulation of the R-equations gives the "direct solution" developed by Stuelpnagel as well as a solution based on the singular value decomposition of B.

Some algorithms become singular when the B matrix is singular (as is always the case when $n = 2$). Mortari offers a general method to eliminate the singularity for the case $n = 2$. When $n = 2$, it is possible to add the unit vector $\mathbf{s}_3 = (\mathbf{s}_1 \times \mathbf{s}_2)/\sin(\vartheta_s)$ and the associated vector $\mathbf{v}_3 = (\mathbf{v}_1 \times \mathbf{v}_2)/\sin(\vartheta_v)$ to the data without affecting the computed optimal attitude. So $\mathbf{s}_3$ and $\mathbf{v}_3$ can be added to the data to avoid a singularity. New weights β_1 and β_2 that replace α_1 and α_2 are such that $\beta_1/\beta_2 = \alpha_1/\alpha_2$ and $\beta_1 + \beta_2 + \beta_3 = 1$. Mortari suggests the values $\beta_1 = 2\alpha_1/3$, $\beta_2 = 2\alpha_2/3$, and $\beta_3 = 1/3$ because these values maximize the distance from the singularity provided by the value of $|\det(B)|$.

2.9 Singular Value Decomposition Algorithm

In [10] a singular value decomposition approach is used to solve Wahba's problem by finding the matrix A_{opt} that minimizes

$$L(A) = \frac{1}{2}\sum_{i=1}^{n} a_i \|\mathbf{b}_i - A\mathbf{r}_i\|^2 \tag{2.66}$$

where $\mathbf{r}_i$ are inertial frame unit vectors, $\mathbf{b}_i$ are the corresponding unit vectors in the body frame, a_i are weights, and n is the number of observations. The weights are normalized so that

$$\sum_{i=1}^{n} a_i = 1. \tag{2.67}$$

Then

$$\begin{aligned}
L(A) &= \frac{1}{2}\sum_{i=1}^{n} a_i(\mathbf{b}_i - A\mathbf{r}_i)^T(\mathbf{b}_i - A\mathbf{r}_i) \\
&= \frac{1}{2}\sum_{i=1}^{n} a_i(\mathbf{b}_i^T - \mathbf{r}_i^T A^T)(\mathbf{b}_i - A\mathbf{r}_i) \\
&= \frac{1}{2}\sum_{i=1}^{n} a_i[\mathbf{b}_i^T - 2\mathbf{b}_i^T A\mathbf{r}_i + \mathbf{r}_i^T A^T A\mathbf{r}_i] \\
&= \frac{1}{2}\sum_{i=1}^{n} a_i[2 - 2\mathbf{b}_i^T A\mathbf{r}_i] \\
&= \sum_{i=1}^{n} a_i - \sum_{i=1}^{n} \mathbf{b}_i^T A\mathbf{r}_i \\
&= 1 - \sum_{i=1}^{n} \mathbf{b}_i^T A\mathbf{r}_i.
\end{aligned}$$

Then the cost function can then be written as

$$L(A) = 1 - \mathrm{tr}(AB^T) \tag{2.69}$$

where

$$B = \sum_{i=1}^{n} a_i \mathbf{b}_i \mathbf{r}_i^T. \tag{2.70}$$

The singular value decomposition of the matrix B is given by

$$B = USV^T$$

where U and V are orthogonal matrices and

$$S = \mathrm{diag}(s_1, s_2, s_3) \quad s_1 \geq s_2 \geq s_3 \geq 0.$$

Define

$$
\begin{aligned}
U_+ &= U[\mathrm{diag}(1, 1, \det(U))] \\
V_+ &= V[\mathrm{diag}(1, 1, \det(V))]
\end{aligned}
$$

and

$$W = U_+^T A V_+ = \cos(\Phi)I + (1 - \cos(\Phi))\mathbf{e}\mathbf{e}^T - \sin(\Phi)[e^\times]$$

where

$$[e^\times] = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix}.$$

Thus W can be represented by a Euler axis/angle rotation with unit vector $\mathbf{e}$ and rotation angle Φ. Define

$$
\begin{aligned}
S' &= \mathrm{diag}(s_1, s_2, ds_3) && (2.71) \\
d &= \det(U)\det(V) = \pm 1. && (2.72)
\end{aligned}
$$

Then B can be written as

$$B = U_+ S' V_+^T. \qquad (2.73)$$

Substitute Equation (2.73) into Equation (2.69) to get

$$
\begin{aligned}
L(A) &= 1 - \mathrm{tr}(S'W) \\
&= 1 - \cos(\Phi)\,\mathrm{tr}(S') - (1 - \cos(\Phi))[s_1 e_1^2 + s_2 e_2^2 + ds_3 e_3^2] + \mathrm{tr}(S') - \mathrm{tr}(S') \\
&= 1 - \mathrm{tr}(S') + (1 - \cos(\Phi))[s_1 + s_2 + ds_3 - s_1 e_1^2 - s_2 e_2^2 - ds_3 e_3^2] \\
&= 1 - \mathrm{tr}(S') + (1 - \cos(\Phi))[s_1(e_1^2 + e_2^2 + e_3^3) + s_2 + ds_3 - s_1 e_1^2 - s_2 e_2^2 - ds_3 e_3^2] \\
&= 1 - \mathrm{tr}(S') + (1 + \cos(\Phi))[s_2 + ds_3 + (s_1 - s_2)e_2^2 + (s_1 - ds_3)e_3^2].
\end{aligned}
$$

$L(A)$ is minimized when $\Phi = 0$, which gives $W = I$ so

$$L(A_{opt}) = 1 - \text{tr}(S') = 1 - s_1 - s_2 - ds_3$$

and

$$A_{opt} = U_+ V_+^T = U[\text{diag}(1, 1, d)]V^T. \tag{2.74}$$

The minimum is unique unless $s_2 + ds_3 = 0$, in which case there is a family of minimizing W matrices given by setting $\mathbf{e}_2 = \mathbf{e}_3 = 0$

$$W = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\Phi) & \sin(\Phi) \\ 0 & -\sin(\Phi) & \cos(\Phi) \end{bmatrix}.$$

Equation (2.74) is a transformation from the inertial frame to the body frame via two transformations. The matrix V_+^T transforms from the inertial frame to an intermediate from (S-frame), and U_+ transforms from the S-frame to the body frame. The rank of B determines the solution's uniqueness. If B has rank less than two, the solution is not unique. The sensitivity in the attitude as a function of the variations $\delta \mathbf{r}_i$ and $\delta \mathbf{b}_i$ is given by

$$z = \sum_{i=1}^{n} a_i[(U_+^T \delta \mathbf{b}_i) \times (V_+^T \mathbf{r}_i) + (U_+^T \mathbf{b}_i) \times (V_+^T \delta \mathbf{r}_i)].$$

In summary the algorithm is as follows:

1. Compute B from Equation (2.70).

2. Find the singular value decomposition of B.

3. Compute d from Equation (2.72).

4. Compute A_{opt} from Equation (2.74).

5. Compute $L(A_{opt})$ and any desired statistics.

2.10 Fast Optimal Attitude Matrix (FOAM) Algorithm

In [11] Markley develops the FOAM algorithm to solve Wahba's problem. Markley presents Wahba's loss function as detailed in §2.9. The cost function can be rewritten as

$$\begin{aligned} L(A) &= \lambda_0 - \operatorname{tr}(AB^T) \\ \lambda_0 &= \sum_{i=1}^{n} a_i \\ B &= \sum_{i=1}^{n} a_i \mathbf{b}_i \mathbf{r}_i^T . \end{aligned}$$

The orthogonal matrix A that maximizes $\operatorname{tr}(AB^T)$ minimizes the expression

$$\|A - B\|^2 = \operatorname{tr}[(A-B)(A-B)^T] = \operatorname{tr}(I) - 2\operatorname{tr}(AB^T) + \|B\|^2$$

so Wahba's problem is equivalent to finding the orthogonal matrix A that is closest to B in Euclidean norm.

The matrix B has the decomposition

$$B = U_+ \operatorname{diag}[S_1, S_2, S_3] V_+^T$$

where U_+ and V_+ are orthogonal matrices required to have positive det

$$S_1 \geq S_2 \geq |S_3|.$$

The optimal attitude estimate is then

$$A_{opt} = U_+ V_+^T .$$

Matrix B has the properties

$$\begin{aligned}
\|B\|_F^2 &= S_1^2 + S_2^2 + S_3^2 \\
\det(B) &= \det(U_+ \operatorname{diag}[S_1, S_2, S_3] V_+^T) \\
&= \det(U_+) \det(\operatorname{diag}[S_1, S_2, S_3]) \det(V_+^T) \\
&= (1)(S_1 S_2 S_3)(1) \\
&= S_1 S_2 S_3 \\
\operatorname{adj}(B^T) &= U_+ \operatorname{diag}[S_2 S_3, S_3 S_1, S_1 S_2] V_+^T \\
B B^T B &= U_+ \operatorname{diag}[S_1^3, S_2^3, S_3^3] V_+^T
\end{aligned}$$

all of which can be evaluated without computing the SVD. They are used to compute A_{opt}

$$A_{opt} = [(\kappa + \|B\|^2)B + \lambda \operatorname{adj}(B^T) - B B^T B]/\zeta \tag{2.75}$$

$$\kappa = S_2 S_3 + S_3 S_1 + S_1 S_2 \tag{2.76}$$

$$\lambda = S_1 + S_2 + S_3 \tag{2.77}$$

$$\zeta = (S_2 + S_3)(S_3 + S_1)(S_1 + S_2). \tag{2.78}$$

The SVD is not necessary to calculate the coefficients κ, λ, and ζ. The coefficients κ and ζ can be written in terms of λ

$$\kappa = \frac{1}{2}(\lambda^2 - \|B\|^2) \tag{2.79}$$

$$\zeta = \kappa\lambda - \det(B) \tag{2.80}$$

so $A(\lambda) = A_{opt}$ when $\lambda = S_1 + S_2 + S_3$. λ can be found with the quartic polynomial

$$0 = p(\lambda) = (\lambda^2 - \|B\|^2)^2 - 8\lambda \det(B) - 4\|\operatorname{adj}(B)\|^2.$$

The roots are all real, and the maximum root is required (it is unique unless $S_2 + S_3 = 0$ in which case the attitude solution is not unique). λ can be found by applying Newton's method.

The covariance matrix P for A_{opt} can be given as

$$P = \lambda_0 \sigma_{tot}^2 (\kappa I + BB^T)/\zeta \tag{2.81}$$

where

$$\sigma_{tot}^2 = \left(\sum_{i=1}^{n} \sigma_i^{-2} \right)^{-1} . \tag{2.82}$$

The i^{th} measurement error vector components are assumed to have a Gaussian distribution with respect to the actual vector, and phase is assumed to have a uniform distribution with variance σ_i^2 per axis. The two choices for λ_0 are 1 and σ_{tot}^{-2}. The normalized ($\lambda_0 = 1$) form is useful with fixed-point arithmetic or if measurement weights are arbitrarily assigned. The unnormalized form ($\lambda_0 = \sigma_{tot}^{-2}$) is good if weights are computed with measurement variances.

2.11 Alternative Quaternion Attitude Estimation Algorithm

In [12] Markley presents another method for attitude estimation based on Wahba's problem. This method also uses SVD the formulation of Wahba's problem presented in §2.9. The loss function can be rewritten as

$$\begin{aligned} L(A) &= \lambda_0 - \mathrm{tr}(AB^T) && (2.83)\\ \lambda_0 &= \sum_{i=1}^{n} a_i && (2.84)\\ B &= \sum_{i=1}^{n} a_i \mathbf{b}_i \mathbf{r}_i^T . && (2.85) \end{aligned}$$

The assumed orthogonality of A gives

$$\|A - B\|^2 = \mathrm{tr}[(A - B)(A - B)^T] = 3 - 2\,\mathrm{tr}(AB^T) + \|B\|^2.$$

This norm is minimized by the same matrix that maximizes $\mathrm{tr}(AB^T)$.

Let

$$\zeta(\lambda, B) = \frac{1}{2}\lambda(\lambda^2 - \|B\|^2) - \det(B).$$

Then the optimal matrix A_{opt} is given by

$$A(\lambda) = \left[\frac{1}{2}(\lambda^2 + \|B\|^2)B + \lambda\,\mathrm{adj}(B^T) - BB^TB\right]/\zeta(\lambda, B) \tag{2.86}$$

where λ is the largest root of the quartic equation resulting from

$$\lambda = \mathrm{tr}[A(\lambda)B^T].$$

Several methods exist to solve for A_{opt} that involve the computation of λ. λ should be close to the value of λ_0 from Equation (2.84) since

$$L(A_{opt}) = \lambda_0 - \lambda \geq 0,$$

and with small measurement errors the loss function should be close to zero. The resulting attitude estimate is

$$\begin{aligned} A_0 &= M/\zeta(\lambda_0, B) & (2.87) \\ M &= \frac{1}{2}(\lambda_0^2 + \|B\|^2)B + \lambda_0\,\mathrm{adj}(B^T) - BB^TB. & (2.88) \end{aligned}$$

The estimate A_0 is only approximately orthogonal.

A variant of Shepperds' algorithm can be used to obtain the normalized attitude quaternion q from A_0 to construct an orthogonal attitude matrix A_{est}

$$A_{est} = \begin{bmatrix} q_4^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_2q_1 - q_3q_4) & q_4^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_3q_1 + q_2q_4) & 2(q_3q_2 - q_1q_4) & q_4^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}. \tag{2.89}$$

If i, j, k is a cyclic permutation of 1, 2, 3, the quaternion components obey the relations

$$\begin{aligned}
4\zeta(\lambda_0, B)q_i^2 &\approx \zeta(\lambda_0, B) + M_{ii} - M_{jj} - M_{kk} = v_i \\
4\zeta(\lambda_0, B)q_iq_j &\approx M_{ij} + M_{ji} = v_j \\
4\zeta(\lambda_0, B)q_iq_k &\approx M_{ik} + M_{ki} = v_k \\
4\zeta(\lambda_0, B)q_iq_4 &\approx M_{jk} - M_{kj} = v_4 = w_i \\
4\zeta(\lambda_0, B)q_jq_4 &\approx M_{ki} - M_{ik} = w_j \\
4\zeta(\lambda_0, B)q_kq_4 &\approx M_{ij} - M_{ji} = w_k \\
4\zeta(\lambda_0, B)q_4^2 &\approx \zeta(\lambda_0, B) + M_{ii} + M_{jj} + M_{kk} = w_4.
\end{aligned}$$

Let i correspond to the index of the largest diagonal element of M, and define the quaternion components for $l = 1, \ldots, 4$ by

$$q_l = v_l/\|v\| \quad \text{for} \quad M_{jj} + M_{kk} < 0$$

or

$$q_l = w_l/\|w\| \quad \text{for} \quad M_{jj} + Mkk \geq 0$$

where $\|v\|$ and $\|w\|$ are the Euclidean norms of the v and w.

2.12 Euler-q Algorithm

In [13] Mortari develops the Euler-q algorithm. It is desired to find the attitude matrix A that rotates unit vectors $\mathbf{v}_i$ from the inertial frame to corresponding unit vectors $\mathbf{s}_i$ in the body frame. A can be expressed in terms of the Euler axis $\mathbf{e}$, a unit 3-vector about which a rotation takes place, and the the Euler angle ϕ, the angle of rotation about the Euler axis, as follows

$$A = I\cos(\phi) + (1 - \cos(\phi))\mathbf{e}\mathbf{e}^T - \tilde{e}\sin(\phi)$$

where $\tilde{e}$ is the 3×3 skew-symmetric, cross-product matrix

$$\tilde{e} = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix}.$$

Let β_i represent the accuracy of the i^{th} sensor. Then the sensor relative precision can be defined as

$$\alpha_i = 1/\left(\beta_i \sum_{k=1}^{n} (1/\beta_k)\right).$$

Wahba's problem is to find the 3×3 matrix A that minimizes the loss function

$$L_W(A) = \frac{1}{2}\sum_{i=1}^{n} \alpha_i \|\mathbf{s}_i - A\mathbf{v}_i\|^2 = 1 - \sum_{i=1}^{n} \alpha_i \mathbf{s}_i^T A \mathbf{v}_i$$

or equivalently maximizes the gain function

$$\begin{aligned} G_W(A) &= 1 - L_W(A) = \sum_{i=1}^{n} \alpha_i \mathbf{s}_i^T A \mathbf{v}_i = \operatorname{tr}(AB^T) \\ B &= \sum_{i=1}^{n} \alpha_i \mathbf{s}_i \mathbf{v}_i^T. \end{aligned}$$

The gain function can be rewritten as

$$G_W(\mathbf{e}, \phi) = \cos(\phi)\operatorname{tr}(B) + (1 - \cos(\phi))\mathbf{e}^T B \mathbf{e} + \sin(\phi)\mathbf{z}^T \mathbf{e}$$

$$\mathbf{z} = \sum_{i=1}^{n} \alpha_i \mathbf{s}_i \times \mathbf{v}_i.$$

Define a loss function

$$
\begin{aligned}
L_M(\mathbf{e}) &= \sum_{i=1}^{n} \xi_i \delta_i^2 = \sum_{i=1}^{n} \xi_i \mathbf{e}^T \mathbf{d}_i \mathbf{d}_i^T \mathbf{e} = \mathbf{e}^T H \mathbf{e} \\
\mathbf{d}_i &= \mathbf{v}_i - \mathbf{s}_i / \|\mathbf{v}_i - \mathbf{s}_i\| \\
\delta_i &= \mathbf{e}^T \mathbf{d}_i = \mathbf{d}_i^T \mathbf{e} \\
H &= H^T = \sum_{i=1}^{n} \xi_i \mathbf{d}_i \mathbf{d}_i^T
\end{aligned}
$$

where ξ_i are relative weights. The worst case for the $\mathbf{d}_i$ direction deviation vector occurs at the angle β_i^* when $\mathbf{S}_i$ is displaced from the measured $\mathbf{s}_i$ by the angle β_i and the spherical triangle is right. The value of β_i^* is obtained from

$$\sin(\omega_i)\sin(\beta_i^*) = \sin(\beta_i)$$

where ω_i is the angle between the $\mathbf{s}_i$ and $\mathbf{v}_i$ directions. The relative weights ξ_i are then derived from β_i^* in the same way that α_i are obtained from β_i

$$\xi_i = 1 / \left(\beta_i^* \sum_{k=1}^{n} (1/\beta_k^*) \right).$$

The augmented cost function is defined as

$$L_M^*(\mathbf{e}) = \mathbf{e}^T H \mathbf{e} - \lambda(\mathbf{e}^T \mathbf{e} - 1)$$

which leads to

$$H\mathbf{e} = \lambda \mathbf{e}.$$

Thus, the Euler axis is the eigenvector of the H matrix associated with the eigenvalue λ so

$$L_M(\mathbf{e}) = \lambda = \mathbf{e}^T H \mathbf{e}.$$

Since $L_M(\mathbf{e})$ has to be a minimum, the unknown eigenvalue has to be the smallest λ

$$H\mathbf{e}_{opt} = \lambda_{min}\mathbf{e}_{opt}.$$

λ_{min} can be found using the characteristic equation of H

$$\begin{aligned}
&0 = \lambda^3 + a\lambda^2 + b\lambda + c \\
&a = -\operatorname{tr}(H) \\
&b = \operatorname{tr}(\operatorname{adj}(H)) \\
&c = -\det(H).
\end{aligned}$$

Since H is symmetric positive semidefinite, its eigenvalues are real and nonnegative. Let $0 \le \lambda_1 \le \lambda_2 \le \lambda_3$, and define the variables

$$\begin{aligned}
p^2 &= (a/3)^2 - (b/3) \\
q &= [(b/2) - (a/3)^2](a/3) - (c/2) \\
w &= \frac{1}{3}\cos^{-1}(q/p^3)
\end{aligned}$$

which leads to

$$\begin{aligned}
&\lambda_1 = -p(\sqrt{3}\sin(w) + \cos(w)) - a/3 \\
&\lambda_2 = p(\sqrt{3}\sin(w) - \cos(w)) - a/3 \\
&\lambda_3 = 2p\cos(w) - a/3.
\end{aligned}$$

The calculation of $\mathbf{e}_{opt}$ proceeds as follows

$$(H - \lambda_{min}I)\mathbf{e}_{opt} = M\mathbf{e}_{opt} = \begin{bmatrix} m_1^T \\ m_2^T \\ m_3^T \end{bmatrix} \mathbf{e}_{opt} = \begin{bmatrix} m_a & m_x & m_y \\ m_x & m_b & m_z \\ m_y & m_z & m_c \end{bmatrix} \mathbf{e}_{opt} = 0.$$

The direction of the optimal Euler axis can be found from the cross product of any two row vectors of M

$$\begin{aligned}\mathbf{e}_1 &= \mathbf{m}_2 \times \mathbf{m}_3\\ \mathbf{e}_2 &= \mathbf{m}_3 \times \mathbf{m}_1\\ \mathbf{e}_3 &= \mathbf{m}_1 \times \mathbf{m}_2.\end{aligned}$$

The most accurate $\mathbf{e}_i$ is the one with the greatest modulus, which is determined by finding the greatest p_i

$$\begin{aligned}p_1 &= |m_b m_c - m_z^2|\\ p_2 &= |m_a m_c - m_y^2|\\ p_3 &= |m_a m_b - m_x^2|.\end{aligned}$$

Let $p_k = \max(p_1, p_2, p_3)$. Then Euler axis with the greatest modulus is

$$\mathbf{e}_{opt} = \mathbf{e}_k / \|\mathbf{e}_k\|.$$

The optimal Euler angle can be derived from

$$\begin{aligned}\sin(\phi_{opt}) &= (1/D)\mathbf{z}^T \mathbf{e}_{opt}\\ \cos(\phi_{opt}) &= (1/D)(\mathrm{tr}(B) - \mathbf{e}_{opt}^T B \mathbf{e}_{opt})\\ D^2 &= (\mathbf{z}^T \mathbf{e}_{opt})^2 + (\mathrm{tr}(B) - \mathbf{e}_{opt}^T B \mathbf{e}_{opt})^2.\end{aligned}$$

The matrix M is singular or ill-conditioned under one of three conditions: a) if the row vectors of M are parallel (colinear), b) the Euler axis $\mathbf{e}$ and all observed vectors are approximately coplanar (one vector is nearly linearly dependent on the other two), or c) the row vectors of M become 0. For case (a) there is no solution, but for cases (b) and (c) the method of sequential rotations (MSR) can be applied. MSR states that if an optimal attitude matrix exists for the n unit vector pairs

$(\mathbf{s}_i, \mathbf{v}_i)$, then there exists n unit vector pairs $(\mathbf{s}_i, \mathbf{w}_i)$ that imply an optimal attitude matrix F. The vectors $\mathbf{s}_i$ and $\mathbf{w}_i$ are related by any rotation matrix R: $\mathbf{w}_i = R\mathbf{s}_i$. A is related to F by $F = \begin{bmatrix} \mathbf{f}_1 & \mathbf{f}_2 & \mathbf{f}_3 \end{bmatrix} = AR^T$. If the $(\mathbf{s}_i, \mathbf{v}_i)$ data set implies a singularity, the set $(\mathbf{s}_i, \mathbf{w}_i)$ would not necessarily imply a singularity. So MSR evaluates the attitude of F by using the rotated unit vector $\mathbf{w}_i$ and then computes A as $A = FR$. By using one of the following rotation matrices

$$R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$
$$R_2 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$
$$R_3 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

no new computation is required, only sign changes.

2.13 ESOQ Algorithm

In [14] Mortari presents the commonly used EStimator of the Optimal Quaternion (ESOQ) algorithm. The ESOQ algorithm is a singularity-free algorithm that finds a closed form expression for λ_{max} and q_{opt} of the q-method equation (Equation (2.94) below).

Let $\mathbf{s}_i$ be unit vectors measured in the body frame, $\mathbf{v}_i$ be the actual vectors in the inertial frame, and A be the matrix that rotates $\mathbf{v}_i$ to the body frame. Let β_i be the precision of the i^{th} sensor such that the angle between the true and observed direction is smaller than β_i. The sensor relative precision is defined as

$$\alpha_i = 1/\left(\beta_i \sum_{k=1}^{n} (1/\beta_k)\right). \tag{2.90}$$

Wahba's problem can be solved by computation of the 3×3 rotation matrix A that minimizes

$$L_W(A) = \frac{1}{2}\sum_{i=1}^{n} \alpha_i \|\mathbf{s}_i - A\mathbf{v}_i\|^2 = 1 - \sum_{i=1}^{n} \alpha_i \mathbf{s}_i^T A \mathbf{v}_i \tag{2.91}$$

or equivalently maximizes the gain function

$$\begin{aligned} G_W(A) &= 1 - L_W(A) = \sum_{i=1}^{n} \alpha_i \mathbf{s}_i^T A \mathbf{v}_i = \operatorname{tr}(AB^T) && (2.92) \\ B &= \sum_{i=1}^{n} \alpha_i \mathbf{s}_i \mathbf{v}_i^T. && (2.93) \end{aligned}$$

The q-method solution equation is

$$K q_{opt} = \lambda_{max} q_{opt} \tag{2.94}$$

meaning q_{opt} is the eigenvector of the symmetric matrix K

$$\begin{aligned} K &= \begin{bmatrix} S & \mathbf{z} \\ \mathbf{z}^T & t \end{bmatrix} = \begin{bmatrix} B + B^T - I\operatorname{tr}(B) & \mathbf{z} \\ \mathbf{z}^T & \operatorname{tr}(B) \end{bmatrix} && (2.95) \\ \mathbf{z} &= \sum_{i=1}^{n} \alpha_i \mathbf{s}_i \times \mathbf{v}_i. && (2.96) \end{aligned}$$

associated with its maximum eigenvalue assuming that the vectors $\mathbf{s}_i$ and $\mathbf{v}_i$ are normalized. The characteristic equation of the matrix K can be written as

$$\begin{aligned} 0 &= \lambda^4 + a\lambda^3 + b\lambda^2 + c\lambda + d && (2.97) \\ a &= \operatorname{tr}(K) = 0 && (2.98) \\ b &= -2(\operatorname{tr}(B))^2 + \operatorname{tr}(\operatorname{adj}(B + B^T)) - \mathbf{z}^T\mathbf{z} && (2.99) \\ c &= -\operatorname{tr}(\operatorname{adj}(K)) && (2.100) \\ d &= \det(K). && (2.101) \end{aligned}$$

The third order auxiliary equation of the characteristic equation is

$$u^3 - bu^2 - 4du + 4bd - c^2 = 0 \tag{2.102}$$

which has the solution

$$\begin{aligned} u_1 &= 2\sqrt{p}\cos\left(\frac{1}{3}\cos^{-1}(q/p^{3/2})\right) + b/3 & (2.103)\\ p &= (b/3)^2 + 4d/3 & (2.104)\\ q &= (b/3)^3 - 4db/3 + c^2/2. & (2.105)\end{aligned}$$

Then

$$\begin{aligned} \lambda_1 &= \frac{1}{2}\left(-g_1 - \sqrt{-u_1 - b + g_2}\right) & (2.106)\\ \lambda_2 &= \frac{1}{2}\left(-g_1 + \sqrt{-u_1 - b + g_2}\right) & (2.107)\\ \lambda_3 &= \frac{1}{2}\left(g_1 - \sqrt{-u_1 - b - g_2}\right) & (2.108)\\ \lambda_4 &= \frac{1}{2}\left(g_1 + \sqrt{-u_1 - b - g_2}\right) & (2.109)\\ g_1 &= \sqrt{u_1 - b} & (2.110)\\ g_2 &= 2\sqrt{u_1^2 - 4d} & (2.111)\end{aligned}$$

where

$$-1 \leq \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \lambda_4 = \lambda_{max} \leq 1. \tag{2.112}$$

If $n = 2$

$$\begin{aligned} \lambda_4 &= -\lambda_1 = (g_3 + g_4)/2 & (2.113)\\ \lambda_3 &= -\lambda_2 = (g_3 - g_4)/2 & (2.114)\\ g_3 &= \sqrt{2\sqrt{d} - b} & (2.115)\\ g_4 &= \sqrt{-2\sqrt{d} - b}. & (2.116)\end{aligned}$$

The q-method solution equation is equivalent to

$$(K - \lambda_{max} I) q_{opt} = H q_{opt} = 0 \tag{2.117}$$

which means q_{opt} is perpendicular to each row vector $\mathbf{h}_i$ of symmetric H. Four different and parallel cross-products q_k can be evaluated using the 4-D cross-product

$$q_k(i) = (-1)^{k+i} \det(H_{ki})$$

where $i = 1, \ldots, 4$ and the 3×3 submatrices H_{ki} are obtained from matrix H by deleting the k^{th} row and the i^{th} column. The optimal q is then

$$q_{opt} = q_G / \|q_G\|$$

where G is the index associated with the element $q_k(k)$ with the largest magnitude.

2.14 ESOQ2 Algorithm

In [15] Mortari presents the ESOQ2 algorithm, an Euler angle variation of the ESOQ algorithm. The same procedure as outlined in §2.13 in Equations (2.90)-(2.117) is used to develop the ESOQ2 algorithm.

The quaternion solution to the q-method equation (Equation (2.94)) can be expressed in terms of an Euler axis and angle

$$q = \{\mathbf{e}^T \sin(\phi/2) \;\; \cos(\phi/2)\}^T.$$

It can be shown that the equation for determining the optimal Euler axis is

$$[(t - \lambda_{max})(S - \lambda_{max} I) - \mathbf{z}\mathbf{z}^T]\mathbf{e} = M\mathbf{e} = 0$$

where

$$M = \begin{bmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \mathbf{m}_3^T \end{bmatrix} = \begin{bmatrix} m_a & m_x & m_y \\ m_x & m_b & m_z \\ m_y & m_z & m_c \end{bmatrix}.$$

All row vectors of M are perpendicular to $\mathbf{e}$. The optimal Euler axis $\mathbf{e}$ can be found by taking the cross-product of two rows of M. This fails in two cases:

1. When the rows of M become zero which occurs if $\phi \to 0$ (the Euler angle singularity).

2. When the rows of M become parallel (an unresolvable case).

The three choices for $\mathbf{e}$ are

$$\mathbf{e}_1 = \mathbf{m}_2 \times \mathbf{m}_3$$
$$\mathbf{e}_2 = \mathbf{m}_3 \times \mathbf{m}_1$$
$$\mathbf{e}_3 = \mathbf{m}_1 \times \mathbf{m}_2$$

where the $\mathbf{e}_i$'s differ only in modulus. The most accurate $\mathbf{e}_i$ is the one with the greatest modulus, which is determined by finding the greatest p_i

$$p_1 = |m_b m_c - m_z^2|$$
$$p_2 = |m_a m_c - m_y^2|$$
$$p_3 = |m_a m_b - m_x^2|.$$

Let p_k have the largest value. Then the most reliable Euler axis is

$$\mathbf{e}_{opt} = \mathbf{e}_k / \|\mathbf{e}_k\|.$$

The optimal Euler angle is then derived from

$$X(k) = h\sin(\phi/2) \tag{2.118}$$

$$Y(k) = h\cos(\phi/2) \tag{2.119}$$

where h is a proportional constant and k identifies the element of vector X with the greatest modulus. The optimal quaternion can then be computed from $\mathbf{e}$ and ϕ. To avoid the singularity that occurs when $\phi \to 0$, the MSR can be applied.

2.15 A Slightly Sub-Optimal Algorithm

In [16] an algorithm for the solution of Wahba's problem that requires no iterations and little computation but results in a slightly nonorthogonal matrix solution is presented. Wahba's loss function can be written as

$$L(A) = \frac{1}{2}\sum_{j=1}^{n} \|\mathbf{v}_j - A\mathbf{u}_j\|^2$$

where $\mathbf{u}_j$ are unit vectors in the inertial frame of n observations and $\mathbf{v}_j$ are the corresponding unit vectors in the body frame. The loss function can be generalized by using an $n \times n$ symmetric positive-definite weight matrix W

$$\begin{aligned} L(A) &= \frac{1}{2}\operatorname{tr}(W(AU-V)^T(AU-V)) \\ U &= [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n] \\ V &= [\mathbf{v}_1, v_2, \ldots, \mathbf{v}_n]. \end{aligned}$$

The loss function can be written as

$$\begin{aligned} L(A) &= \frac{1}{2}\operatorname{tr}(W(U^TU + V^TV)) - \operatorname{tr}(AB^T) \\ B &= VWU^T. \end{aligned}$$

The loss function is minimized when $\operatorname{tr}(AB^T)$ is maximized. This occurs when A is the orthogonal matrix that is closest to B in the Euclidean (or Frobenius) norm. The loss function can be further generalized to

$$L(A) = \frac{1}{2}\operatorname{tr}(W(AU - V)^T Z(AU - V))$$

where Z is a 3×3 symmetric positive-definite weight matrix.

Let A_0 be an extremum of the loss function and ϵ an arbitrary variation in the direction of an arbitrary non-zero matrix H. Then

$$\begin{aligned} L(A_0 + \epsilon H) = \frac{1}{2}\|Z^{1/2}(A_0U - V)W^{1/2}\|_F^2 + \epsilon \operatorname{tr}(H^T Z(A_0U - V)WU^T] && (2.120) \\ + \frac{1}{2}\epsilon^2\|Z^{1/2}HUW^{1/2}\|_F^2. && (2.121) \end{aligned}$$

In order for Equation (2.121) have a solution, the following condition must hold

$$(A_0U - V)WU^T = 0.$$

If U is full rank

$$A_0 = VWU^T(UWU^T)^{-1} = B(UWU^T)^{-1}.$$

The solution provides an unbiased estimate of A under some assumptions:

1. Error-free reference vectors

2. Additive random measurement errors

$$\begin{aligned} \mathbf{v}_j = A_{true}\mathbf{u}_j + \mathbf{n}_j \quad \forall j \\ V = A_{true}U + N \end{aligned}$$

3. E[N] = 0.

The attitude estimate is given by

$$\begin{aligned} A_0 &= (A_{true}U + N)WU^T(UWU^T)^{-1} \\ &= A_{true} + \delta A \\ \delta A &= NWU^T(UWU^T)^{-1}. \end{aligned}$$

An estimate of the deviation of A_0 from A_{true} is

$$\begin{aligned} P &= E[(\delta A)^T \delta A] \\ &= (UWU^T)^{-1}UWRWU^T(UWU^T)^{-1} \\ R &= E[N^T N]. \end{aligned}$$

If we choose $W = R^{-1}$, then

$$P = (UWU^T)^{-1} = (UR^{-1}U^T)^{-1}.$$

If $\{\mathbf{n}_j\}$ is a white sequence, then R and W are diagonal matrices.

This method requires a minimum of three observations ($n = 3$). In the unconstrained problem, a third observation can be added (the cross-product $\mathbf{v}_1 \times \mathbf{v}_2$ as a measurement of $\mathbf{u}_1 \times \mathbf{u}_s$) to make U rank 3.

2.16 Conclusions

Many algorithms have been developed to find solutions to Wahba's problem. While many mathematical solutions are offered to solve the problem, Davenport is the first to present a solution in the form of a practical algorithm. Davenport's algorithm develops what has become known as Davenport's equation but does not provide a quick way to solve the equation. However, Davenport's equation is the foundation of many other algorithms. The QUEST algorithm is one such algorithm. Faster than Davenport's method, the QUEST algorithm has the major disadvantage of using only single measurements. Past information is not taken into account so anomalous data can cause bad estimates. To overcome these limitations, variations of the QUEST algorithm have been developed to make the algorithm more like Kalman filtering. These algorithms cannot compute sensor biases and accounting for process noise proves to be too computationally expensive. The REQUEST algorithm makes the QUEST algorithm recursive, but REQUEST only works if high quality sensors are used to get attitude measurements and does not estimate biases. The extended QUEST algorithm is developed to estimate biases but is slow. The energy approach algorithm offers a new formulation of Wahba's problem that is analogous to a physical system; therefore, it is easy to understand. The algorithm has the drawbacks of assuming small errors in measurements and of often encountering singularities when finding a solution. The SVD algorithm, which requires taking a computationally expensive SVD, provides the basis for the FOAM algorithm, which does not require computing an SVD. The FOAM algorithm and most algorithms developed after the FOAM algorithm provide the same solution, but the speeds at which the solution is derived differ. The ESOQ algorithm provides the same optimal solution as the FOAM algorithm but is faster and guaranteed singularity free. The ESOQ2 algorithm differs from the ESOQ algorithm in that it provides a solution in terms of Euler angles rather than a quaternion. The sub-optimal algorithm is computationally inexpensive and the fastest of all the algorithms presented; however, it has the drawback of computing a rotation matrix that is nonorthogonal.

The ESOQ algorithm is best suited for estimating the states of a ballistic rocket in the early stages of flight. This algorithm is fast while providing an orthogonal rotation matrix estimate. As much accuracy in an attitude estimate as possible as quickly as possible is necessary for the system presented in this work. The ESOQ algorithm also accounts for past measurements rather than being a single-point algorithm like other Wahba's problem algorithms. No assumptions are made about the quality of sensors used, which is essential since MEMS sensors are not as accurate as more conventional sensors.

Chapter 3

Magnetometer Navigation

Magnetometers are a key component of solving Wahba's problem as discussed in Chapter 2. In this chapter we present information relevant to the use of a magnetometer for state estimation of a vehicle. In §3.1 we develop equations to model the earth's magnetic field and present a model that implements those equations. Next, in §3.2 we present a model for a magnetometer, a device that measures magnetic fields. We also present several possible error sources that corrupt magnetometer measurements. In §3.3 we present methods to calibrate a magnetometer in order to minimize measurement errors. In §3.4 we present an algorithm that estimates angular rates based on magnetometer measurements. Finally, in §3.5 we evaluate the presented algorithms as they pertain to the problem in this work.

3.1 Modeling Earth's Magnetic Field

To simulate a magnetometer's behavior, a model of the earth's magnetic field is necessary. In [17] Roithmayr develops equations that are necessary to construct such a model.

3.1.1 Equation Development

An infinite series of spherical harmonics can be used to describe magnetic fields in general. Then at a point Q above the earth's surface, the magnetic field vector $\mathbf{B}$ is given by

$$\begin{aligned}\mathbf{B}_{n,m} &= \frac{K_{n,m}a^{n+2}}{R^{n+m+1}}\Big\{\frac{g_{n,m}C_m + h_{n,m}S_m}{R}[(s_\lambda A_{n,m+1} + (n+m+1)A_{n,m})\hat{\mathbf{r}} - A_{n,m+1}\hat{\mathbf{e}}_3] \\ &- mA_{n,m}[(g_{n,m}C_{m-1} + h_{n,m}S_{m-1})\hat{\mathbf{e}}_1 + (h_{n,m}C_{m-1} - g_{n,m}S_{m-1})\hat{\mathbf{e}}_2]\Big\} \qquad (3.1)\end{aligned}$$

where

$$\mathbf{B} = \sum_{n=1}^{\infty}\sum_{m=0}^{n}\mathbf{B}_{n,m}. \qquad (3.2)$$

In Equation (3.1) a is the earth's average radius (6371 km), R is the magnitude of the position vector $\mathbf{R}$ from the earth's center to point Q, $\hat{\mathbf{r}}$ is a unit vector in the direction of $\mathbf{R}$, and $g_{n,m}$ and $h_{n,m}$ are degree n, order m Gauss coefficients. $\hat{\mathbf{e}}_1$, $\hat{\mathbf{e}}_2$, and $\hat{\mathbf{e}}_3$ are unit vectors of an Earth-fixed coordinate system. The coefficients $K_{n,m}$ are determined recursively from either

$$K_{n,m} = \left(\frac{n-m}{n+m}\right)^{1/2} K_{n-1,m} \qquad m = 1, \ldots, \infty,\ n \geq m+1$$

or

$$K_{n,m} = [(n+m)(n-m+1)]^{-1/2} K_{n,m-1} \qquad m = 2, \ldots, \infty,\ n \geq m$$

where $K_{n,0} = 1$ whenever $m = 0$ and $K_{1,1} = 1$. $A_{n,m}$ and $A_{n,m+1}$ are degree n Legendre polynomials (see §B.2) with orders m and $m+1$, respectively. When $n = m$, the polynomial is given by

$$\begin{aligned} A_{n,n} &= (1)(3)(5)\ldots(2n-1) \qquad n = 1, \ldots, \infty \\ &= (2n-1)A_{n-1,n-1} \quad n = 2, \ldots, \infty \end{aligned}$$

and $A_{0,0} = 1$. In general,

$$A_{n,m}(u) = \frac{1}{n-m}[(2n-1)uA_{n-1,m}(u) - (n+m-1)A_{n-2,m}(u)] \quad m = 0, \ldots, \infty,\ n \geq (m+1).$$

Arguments of the Legendre polynomials are $s_\lambda = \sin(\lambda) = \hat{\mathbf{r}} \cdot \hat{\mathbf{e}}_3$ where λ is the geographic latitude. The variables S_m and C_m are defined recursively as

$$\begin{aligned} S_m &= S_1 C_{m-1} + C_1 S_{m-1} \\ C_m &= C_1 C_{m-1} - S_1 S_{m-1} \end{aligned}$$

where

$$\begin{aligned}
S_0 &= 0 \\
S_1 &= \mathbf{R} \cdot \hat{\mathbf{e}}_2 \\
C_0 &= 1 \\
C_1 &= \mathbf{R} \cdot \hat{\mathbf{e}}_1.
\end{aligned}$$

the earth's magnetic field can be modeled as a tilted dipole as long as the point Q is not near the magnetic poles. For the dipole model

$$\begin{aligned}
A_{1,0}(s_\lambda) &= s_\lambda \\
A_{1,1}(s_\lambda) &= 1 \\
A_{1,2}(s_\lambda) &= 0 \\
K_{1,1} &= K_{1,0} = 1.
\end{aligned}$$

Then

$$\begin{aligned}
\mathbf{B}_{1,0} &= \left(\frac{a}{R}\right)^3 g_{1,0}(3s_\lambda\hat{\mathbf{r}} - \hat{\mathbf{e}}_3) \\
\mathbf{B}_{1,1} &= \left(\frac{a}{R}\right)^3 [3(g_{1,1}\hat{\mathbf{r}} \cdot \hat{\mathbf{e}}_1 + h_{1,1}\hat{\mathbf{r}} \cdot \hat{\mathbf{e}}_2)\hat{\mathbf{r}} - (g_{1,1}\hat{\mathbf{e}}_1 + h_{1,1}\hat{\mathbf{e}}_2)].
\end{aligned}$$

So

$$\mathbf{B}_{1,0} + \mathbf{B}_{1,1} = \left(\frac{a}{R}\right)^3 [3(\hat{r} \cdot \mathbf{M})\hat{r} - \mathbf{M}]$$

where the terrestrial dipole moment $\mathbf{M}$ is defined as

$$\mathbf{M} = g_{1,1}\hat{\mathbf{e}}_1 + h_{1,1}\hat{\mathbf{e}}_2 + g_{1,0}\hat{\mathbf{e}}_3.$$

3.1.2 Model Implementation

Two implementations of models of the earth's magnetic field are commonly used: the International Geomagnetic Reference Field (IGRF) [18] and the World Magnetic Model (WMM) (an eighth-order model) [19] which is used by the U.S. DoD and NATO. Due to the changes in the behavior of the earth's magnetic field such as from secular variation (slow changes in time of the main magnetic field), these models are updated every five years (last updated in 2005). Our development is based on the WMM, and further discussion is concentrated solely on the WMM.

the earth's magnetic field is generated mainly by three sources:

1. The main field generated by the earth's outer core ($\mathbf{B}_m$).

2. The crustal field generated by the earth's crust and upper mantle ($\mathbf{B}_c$).

3. Electrical currents in the upper magnetosphere and atmosphere which induce electrical currents in the ground and oceans ($\mathbf{B}_d$).

The total magnetic field can then be written as

$$\mathbf{B}(\mathbf{r}, t) = \mathbf{B}_m(\mathbf{r}, t) + \mathbf{B}_c(\mathbf{r}) + \mathbf{B}_d(\mathbf{r}, t)$$

where $\mathbf{r}$ is a position vector and t is time. To model the earth's magnetic field the WMM uses data gathered from the Danish Ørsted and German CHAMP satellites, which have good global coverage and low noise, as well as data from ground observatories.

The WMM only takes into account the contributions of $\mathbf{B}_m$. This introduces errors into the model since $\mathbf{B}_c$ and $\mathbf{B}_d$ are ignored. The WMM also has other error sources. A magnetic sensor will not match the WMM in all locations on the earth; it may observe spatial and temporal anomalies. Spatial anomalies on land are caused by mountain ranges, ore deposits, geological faults, trains, railroad tracks, power lines, and other such conditions. Disturbances in the atmosphere from ionic activity from space will also cause variations in the magnetic field.

Seven quantities describe the geomagnetic field vector $\mathbf{B}$. These quantities are northerly intensity X, easterly intensity Y, vertical intensity (positive downwards) Z, total intensity F, horizontal intensity H, inclination (or dip) I which is the angle between the horizontal plane and the field vector (measured positive downwards), and declination (or magnetic variation) D which is the horizontal angle between true north and the field vector (measured positive eastwards). These values are calculated as follows

$$\begin{aligned}
H &= \sqrt{X^2 + Y^2} \\
F &= \sqrt{H^2 + Z^2} \\
D &= \arctan(Y, X) \\
I &= \arctan(Z, H) \\
GV &= D - \lambda \quad \text{for } \phi > 55^\circ \\
GV &= D + \lambda \quad \text{for } \phi < -55^\circ
\end{aligned}$$

where GV is grid variation, λ is longitude, and ϕ is latitude. The WMM algorithm is used to compute the magnetic field for a given location and time (h, ϕ, λ, t) (see §D.2 for a more complete discussion of these variables), where h is geodetic altitude, ϕ and λ are geodetic latitude and longitude, and t is the time given in decimal year. The algorithm proceeds as follows:

1. Convert ellipsoidal geodetic coordinates (h, ϕ, λ) to spherical geocentric coordinates (r, ϕ', λ).

2. Determine the Gauss coefficients of degree n and order m for the desired time.

3. Compute the field vector components in geocentric coordinates.

4. Convert the field vector components to the geodetic reference frame.

5. Compute H, F, D, I, and GV.

For a more complete discussion of the algorithm details, see [19].

3.2 Magnetometer Modeling

A magnetometer is a device that measures magnetic field strength and direction and in navigation can be used to obtain heading information. A summary of some basics concerning magnetometers as well as a new off-line calibration technique are given by McClendon in his master's thesis [20]. A brief overview of some points in the thesis follows.

In order to properly use magnetometer data, relationships among reference frames must be established. An inertial frame is an Earth-fixed set of axes with the xy-plane tangent to the earth's surface. The x-axis points to the earth's magnetic north. (Declination angles can be used to reconcile magnetic north and true north.) The y-axis points east, and the z-axis points downward. The heading frame is defined such that it shares its z-axis with the inertial frame z-axis. The heading frame's x- and y-axes are coplanar with the inertial frame's x- and y-axes but may be rotated by some angle ψ, the heading. A body frame is defined such that the x-axis points out the front of the body, the y-axis to the right, and the z-axis down. This reference frame rotates along with the body of the rocket or aircraft. One frame can be rotated into the other using the appropriate set of Givens rotations (rotation matrices or quaternions).

Measurements of the earth's magnetic field can be used to determine vehicle heading

$$\psi = -\arctan\left(\frac{B_y^h}{B_x^h}\right) \tag{3.3}$$

where B_y^h and B_x^h are the y- and x-axis components, respectively, of the magnetic field measured in the heading reference frame. Then body frame measurements of magnetic field intensity $\hat{\mathbf{B}}^b = [\hat{B}_x^b, \hat{B}_y^b, \hat{B}_z^b]^T$ can be rotated to the heading frame via

$$\mathbf{B}_{xyz}^h = \begin{bmatrix} \cos(\phi) & \sin(\phi)\sin(\theta) & \sin(\phi)\cos(\theta) \\ 0 & \cos(\theta) & -\sin(\theta) \\ -\sin(\phi) & \sin(\theta)\cos(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \hat{\mathbf{B}}^b.$$

the earth's magnetic field has both vertical and horizontal components with magnitudes that vary with latitude. The horizontal component, which points north, is parallel to the earth's surface. The vertical component does not affect heading calculations unless the magnetometer has a nonzero roll or pitch angle, in which case coupling between the axes occurs.

Several error sources must be accounted for in order to get an accurate measurement of heading. The first commonly encountered error source is soft iron error. This error occurs when materials (typically ferromagnetic metals such as iron, nickel, or cobalt) near the magnetometer warp existing magnetic fields. Soft iron error is a function of the magnetometer's orientation. Another error source is hard iron error. Hard iron error is interference from objects near the magnetometer that emit their own magnetic field. This error can be corrected through calibration. Yet another error source is misalignment of the magnetometer's measurement axes with body axes of the vehicle on which the magnetometer is mounted. A final error source is scale factor errors in the sensor. These errors can be used to produce the following model for a magnetometer:

$$\hat{\mathbf{B}}^b = K_s K_{si} K_m (\mathbf{B}^b + \delta \mathbf{B}^b). \tag{3.4}$$

$\mathbf{B}^b$ is a 3×1 vector of the true magnetic field in the body frame, K_m is a 3×3 matrix representing measurement transformation from mounting error, K_{si} is a 3×3 matrix that factors in soft iron error, K_s is a 3×3 matrix of sensor scale factor errors, and $\delta \mathbf{B}^b$ is a vector of measurement biases resulting from hard iron error sources. A more complete model includes the addition of η_M, a zero-mean Gaussian noise term with variance σ_M:

$$\hat{\mathbf{B}}^b = K_s K_{si} K_m (\mathbf{B}^b + \delta \mathbf{B}^b) + \eta_M. \tag{3.5}$$

3.3 Magnetometer Calibration

In this section we present some methods for the calibration of a magnetometer. In §3.3.1 we review methods presented by Alonso and Shuster [21] to estimate magnetometer bias when attitude is unknown. Next, we summarize the TWOSTEP algorithm [22] in §3.3.2. Finally, in §3.3.3 we present a recursive least squares method developed by Hodgart and Tortora [23].

3.3.1 Attitude Independent Bias Estimation

In [21] Alonso and Shuster present several algorithms that estimate magnetometer bias without knowledge of attitude. The magnetometer model used is

$$\mathbf{B}_k = A_k\mathbf{H}_k + \mathbf{b} + \epsilon_k, \quad k = 1, \ldots, N \tag{3.6}$$

($\mathbf{B}_k$ is the magnetic field measurement at time t_k, $\mathbf{H}_k$ is value of the geomagnetic field in an Earth-fixed reference frame, A_k is the magnetometer's attitude with respect to the earth-fixed frame, $\mathbf{b}$ is the magnetometer's constant bias, and ϵ_k is measurement noise). Scalar measurements and scalar noise values can be used to estimate bias. Define

$$\begin{aligned} z_k &= \|\mathbf{B}_k\|^2 - \|\mathbf{H}_k\|^2 \\ \nu_k &= 2(\mathbf{B}_k - \mathbf{b}) \cdot \epsilon_k - |\epsilon_k|^2. \end{aligned}$$

Then $A_k\mathbf{H}_k = \mathbf{B}_k - \mathbf{b} - \epsilon_K$. Since A_k is a rotation matrix, it does not change the magnitude of $\mathbf{H}_k$. Thus,

$$\begin{aligned} \|\mathbf{H}_k\|^2 &= \|A_k\mathbf{H}_k\|^2 = (\mathbf{B}_k - \mathbf{b} - \epsilon_k)^T(\mathbf{B}_k - \mathbf{b} - \epsilon_k) \\ &= \|\mathbf{B}_k\|^2 - 2\mathbf{B}_k \cdot \mathbf{b} + \|\mathbf{b}\|^2 + \|\epsilon_k\|^2 - 2\mathbf{B}_k\epsilon_k + 2\mathbf{b}\epsilon_k \\ &= \|\mathbf{B}_k\|^2 - 2\mathbf{B}_k \cdot \mathbf{b} + \|\mathbf{b}\|^2 + \|\epsilon_k\|^2 - 2(\mathbf{B}_k - \mathbf{b}) \cdot \epsilon_k \\ &= \|\mathbf{B}_k\|^2 - 2\mathbf{B}_k \cdot \mathbf{b} + \|\mathbf{b}\|^2 - \nu_k. \end{aligned}$$

So

$$\begin{aligned} z_k &= \|\mathbf{B}_k\|^2 - \|\mathbf{B}_k\|^2 + 2\mathbf{B}_k \cdot \mathbf{b} - \|\mathbf{b}\|^2 + \nu_k \\ &= 2\mathbf{B}_k \cdot \mathbf{b} - \|\mathbf{b}\|^2 + \nu_k, \quad k = 1, \ldots, N. \end{aligned} \tag{3.7}$$

It is assumed that $\epsilon_k \sim \mathcal{N}(0, \Sigma_k)$. Then $\nu_k \sim \mathcal{N}(\mu_k, \sigma_k^2)$ where

$$\begin{aligned}
\mu_k &= E\{\nu_k\} = -\operatorname{tr}(\Sigma_k) \\
\sigma_k^2 &= E\{\nu_k^2\} - \mu_k^2 = 4(\mathbf{B}_k - \mathbf{b})^T \Sigma_k (\mathbf{B}_k - \mathbf{b}) + 2\operatorname{tr}(\Sigma_k^2).
\end{aligned}$$

This model is the basis for the following bias estimators presented.

Scoring

The negative-log-likelihood function for magnetometer bias as modeled by Equation (3.6) is

$$J(\mathbf{b}) = \frac{1}{2} \sum_{k=1}^{N} \left[\frac{1}{\sigma_k^2} (z_k - 2\mathbf{B}_k \cdot \mathbf{b} + \|\mathbf{b}\|^2 - \mu_k)^2 + \log \sigma_k^2 + \log 2\pi \right],$$

which is quartic in $\mathbf{b}$. The maximum-likelihood estimate $\mathbf{b}^*$ for $\mathbf{b}$ satisfies

$$\left.\frac{\partial J}{\partial \mathbf{b}}\right|_{\mathbf{b}^*} = 0.$$

One solution procedure is to use scoring (taking the partial derivative a log-likelihood function with respect to some parameter), which for this case is the Newton-Raphson approximation

$$\begin{aligned}
\mathbf{b}_0^{NR} &= 0 \\
\mathbf{b}_{i+1}^{NR} &= \mathbf{b}_i^{NR} - \left[\frac{\partial^2 J}{\partial \mathbf{b} \partial \mathbf{b}^T} (\mathbf{b}_i^{NR}) \right]^{-1} \frac{\partial J}{\partial \mathbf{b}} (\mathbf{b}_i^{NR})
\end{aligned}$$

where

$$\begin{aligned}
\frac{\partial^2 J}{\partial \mathbf{b} \partial \mathbf{b}^T} &= \sum_{k=1}^{N} \frac{1}{\sigma_k^2} \left[4(\mathbf{B}_k - \mathbf{b})(\mathbf{B}_k - \mathbf{b})^T + 2(z_k - 2\mathbf{B}_k \cdot \mathbf{b} + \|\mathbf{b}\|^2 - \mu_k) I_{3\times3} \right] \\
\frac{\partial J}{\partial \mathbf{b}} &= -\sum_{k=1}^{N} \frac{1}{\sigma_k^2} (z_k - 2\mathbf{B}_k \cdot \mathbf{b} + \|\mathbf{b}\|^2 - \mu_k) 2(\mathbf{B}_k - \mathbf{b}).
\end{aligned}$$

Estimates of bias by scoring can also be obtained with the Gauss-Newton algorithm. The algorithm proceeds as follows

$$\begin{aligned}\mathbf{b}_0^{GN} &= 0 \\ \mathbf{b}_{i+1}^{GN} &= \mathbf{b}_i^{GN} - F_{bb}^{-1}\frac{\partial J}{\partial \mathbf{b}}\left(\mathbf{b}_i^{GN}\right)\end{aligned}$$

where

$$F_{bb} = \sum_{k=1}^{N} \frac{1}{\sigma_k^2} 4(\mathbf{B}_k - \mathbf{b})(\mathbf{B}_k - \mathbf{b})^T.$$

The estimate error covariance is then given by $P_{bb} = F_{bb}^{-1}$. For quadratic functions both the Newton-Raphson and Gauss-Newton methods tend to converge rapidly once the estimate is sufficiently close to the minima. However, for quartic functions the possibility exists to become stuck in local minima rather than converging to the global minimum.

Fixed-Point Method

The fixed-point method proceeds as follows. First, define

$$\begin{aligned}G &= \sum_{k=1}^{N} \frac{1}{\sigma_k^2}[4\mathbf{B}_k\mathbf{B}_k^T + 2(z_k - \mu_k)I_{3\times 3}] \\ \mathbf{a} &= \sum_{k=1}^{N} \frac{1}{\sigma_k^2}(z_k - \mu_k)2\mathbf{B}_k \\ \mathbf{f}(\mathbf{b}) &= \sum_{k=1}^{N} \frac{1}{\sigma_k^2}[4(\mathbf{B}_k \cdot \mathbf{b})\mathbf{b} + 2\|\mathbf{b}\|^2(\mathbf{B}_k - \mathbf{b})].\end{aligned}$$

Then the optimal solution is given by

$$\mathbf{b}^* = G^{-1}[\mathbf{a} + \mathbf{f}(\mathbf{b}^*)].$$

This can be solved iteratively via

$$\begin{aligned}\mathbf{b}_0^{FP} &= 0 \\ \mathbf{b}_{i+1}^{FP} &= G^{-1}[\mathbf{a} + \mathbf{f}(\mathbf{b}_i^{FP})].\end{aligned}$$

Convergence is usually poor.

Davenport's Approximation

Davenport's method gives an estimate of the bias, but the estimate is inconsistent. That is, the mean-squared error of the estimates does not tend to zero as the number of observations increases. The method begins with the approximate cost function

$$J_D(\mathbf{b}) = \frac{1}{2}\sum_{k=1}^{N} \frac{1}{\sigma_k^2}(z_k - 2\mathbf{B}_k \cdot \mathbf{b} + \lambda^2 - \mu_k)^2$$

where λ is a constant. Then

$$\begin{aligned}\mathbf{b}_D^* &= \mathbf{U} + \lambda^2 \mathbf{V} \\ \mathbf{U} &= \left[\sum_{k=1}^{N} \frac{1}{\sigma_k^2} 4\mathbf{B}_k \mathbf{B}_k^T\right]^{-1} \left[\sum_{k=1}^{N} \frac{1}{\sigma_k^2}(z_k - \mu_k) 2\mathbf{B}_k\right] \\ \mathbf{V} &= \left[\sum_{k=1}^{N} \frac{1}{\sigma_k^2} 4\mathbf{B}_k \mathbf{B}_k^T\right]^{-1} \left[\sum_{k=1}^{N} \frac{1}{\sigma_k^2} 2\mathbf{B}_k\right].\end{aligned}$$

Then choose λ:

$$\begin{aligned}\lambda^2 &= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \\ a &= |\mathbf{V}|^2 \\ b &= 2\mathbf{U} \cdot \mathbf{V} - 1 \\ c &= |\mathbf{U}|^2.\end{aligned}$$

Usually, the ± is chosen to be minus.

Acuña's Algorithm

Acuña's algorithm does not rely on a field model. The derived measurement $z_{k,l}$ and effective measurement error $\Delta_{k,l}$ are defined as

$$\begin{aligned} z_{k,l} &= \|\mathbf{B}_k\|^2 - \|\mathbf{B}_l\|^2 \\ &= 2(\mathbf{B}_k - \mathbf{B}_l) \cdot \mathbf{b} + \Delta_{k,l} \\ \Delta_{k,l} &= \|\mathbf{H}_k\|^2 - \|\mathbf{H}_l\|^2 + \nu_k - \nu_l \end{aligned}$$

where k and l are two different time indices. The cost function is then defined as

$$J(\mathbf{b}) = \frac{1}{2} \sum_{k,l}^{\prime} [z_{k,l} - 2(\mathbf{B}_k - \mathbf{B}_l) \cdot \mathbf{b}]^2$$

where the prime indicates that no index occurs more than once. In other words, n individual magnetometer measurements can yield no more than $n/2$ effective measurements. Then the optimal solution is

$$\begin{aligned} \mathbf{b}_A^* &= \left[\sum_{k,l}^{\prime} 4(\mathbf{B}_k - \mathbf{B}_l)(\mathbf{B}_k - \mathbf{B}_l)^T \right]^{-1} \sum_{k,l}^{\prime} 2(\mathbf{B}_k - \mathbf{B}_l) z_{k,l} \\ &= \mathbf{b} + \Delta\mathbf{b}_A \end{aligned}$$

where

$$\Delta\mathbf{b}_A = \left[\sum_{k,l}^{\prime} 4(\mathbf{B}_k - \mathbf{B}_l)(\mathbf{B}_k - \mathbf{B}_l)^T \right]^{-1} \sum_{k,l}^{\prime} 2(\mathbf{B}_k - \mathbf{B}_l) \Delta_{k,l}.$$

The estimation error contains both random and systematic terms. The dominate error source determines how to best construct $z_{k,l}$. For a more detailed discussion of Acuña's algorithm, see [21].

3.3.2 TWOSTEP Algorithm Calibration

The TWOSTEP algorithm is a robust algorithm that is commonly used to estimate magnetometer bias. Full calibration had not been achieved with the TWOSTEP algorithm until recently. In [22] Alonso and Shuster outline how the TWOSTEP algorithm estimates a magnetometer's biases and extend the algorithm for full magnetometer calibration.

Bias Estimate

It is desired to find an estimate of $\mathbf{b}$ where the measurement model is given by Equation (3.6). Then $\mathbf{b}$ can be estimated in the maximum likelihood sense by minimizing the cost function

$$J(\mathbf{b}) = \frac{1}{2}\sum_{k=1}^{N}\left[\frac{1}{\sigma_k^2}(z_k - 2\mathbf{B}_k \cdot \mathbf{b} + \|\mathbf{b}\|^2 - \mu_k)^2 + \log(\sigma_k^2) + \log(2\pi)\right].$$

where z_k is defined by Equation (3.7), and the Gaussian noise $\nu_k \sim \mathcal{N}(\mu_k, \sigma_k^2)$ is

$$\nu_k = 2(\mathbf{B}_k - \mathbf{b} \cdot \epsilon_k - \|\epsilon_k\|^2).$$

The minimization involves a quartic (or fourth degree polynomial) in $\mathbf{b}$ so multiple minima and maxima exist. To find the global minimum reliably, a good starting estimate of $\mathbf{b}$ is needed.

The centering approximation gives a reliable starting estimate of $\mathbf{b}$. For a sequence of variables $X_k,\ k = 1, \ldots, N$, center values are defined as

$$\overline{X} = \overline{\sigma}^2 \sum_{k=1}^{N} \frac{1}{\sigma_k^2} X_k \qquad \frac{1}{\overline{\sigma}^2} = \sum_{k=1}^{N} \frac{1}{\sigma_k^2},$$

and centered values are defined as

$$\tilde{X}_k = X_k - \overline{X}, \quad k = 1, \ldots, N.$$

Then the cost function can be written in terms of centered and center values as

$$J(\mathbf{b}) = \tilde{J}(\mathbf{b}) + \overline{J}(\mathbf{b})$$

where

$$\tilde{J}(\mathbf{b}) = \frac{1}{2}\sum_{k=1}^{N} \frac{1}{\sigma_k^2}(\tilde{z}_k - 2\tilde{\mathbf{B}}_k \cdot \mathbf{b} - \tilde{\mu}_k)^2 + \text{terms independent of } \mathbf{b}$$

and

$$\overline{J}(\mathbf{b}) = \frac{1}{2}\frac{1}{\overline{\sigma}^2}(\overline{z} - 2\overline{\mathbf{B}} \cdot \mathbf{b} + \|\mathbf{b}\|^2 - \overline{\mu})^2 + \text{terms independent of } \mathbf{b}.$$

The TWOSTEP algorithm is then used on the cost function to estimate $\mathbf{b}$. First, the $\mathbf{b}$ that minimizes $\tilde{J}(\mathbf{b})$ is found. The minimum, guaranteed since the function is quadratic in $\mathbf{b}$, is equal to

$$\tilde{\mathbf{b}}^* = \tilde{P}_{bb}\sum_{k=1}^{N} \frac{1}{\sigma_k^2}(\tilde{z}_k - \tilde{\mu}_k)2\tilde{\mathbf{B}}_k$$

where

$$\tilde{P}_{bb} = \tilde{F}_{bb}^{-1} = \left[\sum_{k=1}^{N} \frac{1}{\sigma_k^2} 4\tilde{\mathbf{B}}_k\tilde{\mathbf{B}}_k^T\right]^{-1}.$$

The algorithm then uses $\tilde{\mathbf{b}}^*$ as an initial guess for the Gauss-Newton algorithm, which is applied to the full negative-log-likelihood cost function

$$J(\mathbf{b}) = \frac{1}{2}(\mathbf{b} - \tilde{\mathbf{b}}^*)^T\tilde{P}_{bb}^{-1}(\mathbf{b} - \tilde{\mathbf{b}}^*) + \frac{1}{2\overline{\sigma}^2}(\overline{z} - 2\overline{\mathbf{B}} \cdot \mathbf{b} + \|\mathbf{b}\|^2 - \overline{\mu})^2 + \text{terms independent of } \mathbf{b},$$

to find the final estimate of $\mathbf{b}$. Then the iteration to find the bias estimate is

$$\begin{aligned} \mathbf{b}_1^{GN} &= \tilde{\mathbf{b}}^* \\ \mathbf{b}_{i+1}^{GN} &= \mathbf{b}_i^{GN} - \left[\tilde{P}_{bb}^{-1} + \frac{4}{\overline{\sigma}^2}(\overline{\mathbf{B}} - \mathbf{b}_i^{GN})(\overline{\mathbf{B}} - \mathbf{b}_i^{GN})^T\right]^{-1} \frac{\partial J}{\partial \mathbf{b}}(\mathbf{b}_i^{GN}). \end{aligned}$$

Full Calibration

Alonso and Shuster [22] give a more general model for the magnetic field measured by a magnetometer that includes scale factors and nonorthogonality of measurement axes

$$\mathbf{B}_k = T^{-1}[A_k\mathbf{H}_k + \mathbf{b}' + \epsilon'_k] \quad k = 1, \ldots, N.$$

Through a polar decomposition T can be written as $T = Q(I + D)$ where Q is an orthogonal matrix and D is symmetric. Then $\mathbf{b} = Q^T\mathbf{b}'$ and $\epsilon_k = Q^T\epsilon'_k$. When A_k is unknown, Q cannot be estimated, and $\|H_k\|^2$ becomes

$$\|H_k\|^2 = \|(I + D)\mathbf{B}_k - \mathbf{b} - \epsilon_k\|^2.$$

Therefore, only $\mathbf{b}$ and D can be estimated.

Rewriting the magnetometer model as

$$\mathbf{B}_k = (I + D)^{-1}(Q^T A_k\mathbf{H}_k + \mathbf{b} + \epsilon_k) \quad k = 1, \ldots, N$$

yields

$$\begin{aligned} z_k &= \|\mathbf{B}_k\|^2 - \|\mathbf{H}_k\|^2 \\ &= -\mathbf{B}_k^T(2D + D^2)\mathbf{B}_k + 2\mathbf{B}_k^T(I + D)\mathbf{b} - \|\mathbf{b}\|^2 + \nu_k \\ \nu_k &= 2[(I + D)\mathbf{B}_k - \mathbf{b}] \cdot \epsilon_k - \|\epsilon_k\|^2. \end{aligned}$$

To estimate D and $\mathbf{b}$ define

$$E = 2D + D^2$$

$$\mathbf{c} = (I + D)\mathbf{b}.$$

Then

$$z_k = -\mathbf{B}_k E \mathbf{B}_k + 2\mathbf{B}_k^T \mathbf{c} - \|\mathbf{b}(\mathbf{c}, \mathbf{E})\|^2 + \nu_k$$

where

$$\mathbf{E} = \begin{bmatrix} E_{11} & E_{22} & E_{33} & E_{12} & E_{13} & E_{23} \end{bmatrix}.$$

After writing $\mathbf{B}_k^T E \mathbf{B}_k = K_k \mathbf{E}$ where

$$K_k = \begin{bmatrix} B_{1,k}^2 & B_{2,k}^2 & B_{3,k}^2 & 2B_{1,k}B_{2,k} & 2B_{1,k}B_{3,k} & 2B_{2,k}B_{3,k} \end{bmatrix},$$

z_k can be written as

$$\begin{aligned} z_k &= -K_k \mathbf{E} + 2\mathbf{B}_k^T \mathbf{c} - \|\mathbf{b}(\mathbf{c}, \mathbf{E})\|^2 + \nu_k \\ &= L_k \mathbf{\Theta}' - \|\mathbf{b}(\mathbf{\Theta}')\|^2 + \nu_k \end{aligned}$$

where

$$L_k = \begin{bmatrix} 2\mathbf{B}_k^T & \vdots & -K_k \end{bmatrix} \qquad \mathbf{\Theta}' = \begin{bmatrix} \mathbf{c} \\ \mathbf{E} \end{bmatrix}.$$

Then the center and centered values can be calculated

$$\overline{L}_k = \overline{\sigma}^2 \sum_{k=1}^{N} \frac{1}{\sigma_k^2} L_k, \qquad \tilde{L}_k = L_k - \overline{L}.$$

The center and centered measurements are

$$\begin{aligned} \tilde{z}_k &= \tilde{L}_k \cdot \mathbf{\Theta}' + \tilde{\nu}_k, \quad k = 1, \dots, N \\ \overline{z}_k &= \overline{L} \mathbf{\Theta}' - \|\mathbf{b}(\mathbf{\Theta}')\|^2 + \overline{\nu}, \end{aligned}$$

and

$$\|\mathbf{b}(\mathbf{\Theta}')\|^2 = \mathbf{c}^T (I + D)^{-2} \mathbf{c} = \mathbf{c}^T (I + E)^{-1} \mathbf{c}.$$

The cost function is then

$$\tilde{J}(\mathbf{\Theta}') = \frac{1}{2}\sum_{k=1}^{N}\frac{1}{\sigma_k^2}(\tilde{z}_k - \tilde{L}_k\mathbf{\Theta}' - \tilde{\mu}_k)^2 + \text{terms independent of } \mathbf{\Theta}'$$

where

$$\begin{aligned}\tilde{\mathbf{\Theta}}'^{*} &= \tilde{P}_{\Theta,\Theta}\sum_{k=1}^{N}\frac{1}{\sigma_k^2}(\tilde{z}_k - \tilde{\mu}_k)\tilde{L}_k^T \\ \tilde{P}_{\Theta,\Theta}^{-1} &= \sum_{k=1}^{N}\frac{1}{\sigma_k^2}\tilde{L}_k\tilde{L}_k^T.\end{aligned}$$

The center cost function is

$$\overline{J}(\mathbf{\Theta}') = \frac{1}{2\overline{\sigma}^2}(\overline{z} - \overline{L}\mathbf{\Theta}' + \|\mathbf{b}(\mathbf{\Theta}')\|^2 - \overline{\mu})^2,$$

and the complete cost function is

$$J(\mathbf{\Theta}') = \frac{1}{2}(\mathbf{\Theta}' - \mathbf{\Theta}'^{*})^T\tilde{P}_{\Theta,\Theta}(\mathbf{\Theta}' - \mathbf{\Theta}'^{*}) + \overline{J}(\mathbf{\Theta}').$$

The Gauss-Newton algorithm is applied to find $\mathbf{c}^*$ and $\mathbf{E}^*$ where

$$\begin{aligned}\frac{\partial}{\partial c_m}\|\mathbf{b}(\mathbf{\Theta}')\|^2 &= 2((I+E)^{-1}\mathbf{c})_m \\ \frac{\partial}{\partial E_{m,n}}\|\mathbf{b}(\mathbf{\Theta}')\|^2 &= -(2-\delta_{m,n})((I+E)^{-1}\mathbf{c})_m((I+E)^{-1}\mathbf{c})_n.\end{aligned}$$

$((I+E)^{-1}\mathbf{c})_m$ denotes the m^{th} element of $((I+E)^{-1}\mathbf{c})$, and $\delta_{m,n}$ is the Kronecker delta ($\delta_{m,n} = 1$ when $m = n$ and 0 otherwise). Then E^* may be written as $E^* = USU^T$ where U is an orthogonal matrix and S is diagonal. Then define a matrix W such that $S = 2W + W^2$ where W is diagonal so that

$$w_j = -1 + \sqrt{1 + s_j} \quad j = 1, 2, 3.$$

In general, s_j is much less than one so a solution exists. Then the desired parameters D and $\mathbf{b}$ may be estimated as

$$\begin{aligned} D^* &= UWU^T \\ \mathbf{b}^* &= (I + D^*)^{-1}\mathbf{c}^*. \end{aligned}$$

3.3.3 Recursive Least Squares Method

In [23] Hodgart and Tortora develop a recursive least squares method to calibrate a magnetometer that is onboard a satellite orbiting Earth. The method makes some assumptions. First, the satellite points roughly in a zenith-nadir direction. Second, the satellite slowly spins about the zenith-nadir axis about which it is symmetric.

The magnetometer measurement is modeled as

$$\begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \end{bmatrix} \begin{bmatrix} B_{xn} \\ B_{yn} \\ B_{zn} \\ B_{on} \end{bmatrix}$$

or

$$\mathbf{B} = C\mathbf{B}_n$$

where B is an optimal estimate of the magnetic field, $\mathbf{B}_n = \begin{bmatrix} B_{xn} & B_{yn} & B_{zn} \end{bmatrix}^T$ is a vector of nominal values obtained a priori through on-ground calibration, and B_{on} is an arbitrary offset. Coefficients c_{11}, c_{22}, c_{33}, c_{14}, c_{24}, and c_{34} are magnitude dependent, and the authors cite batch methods to find these values. The remaining coefficients are direction dependent, and a recursive least squares algorithm is used to obtain those values.

Before the algorithm can be performed, several preliminary steps must be taken. Data must be collected to obtain a file of magnetic data for each axis of the magnetometer. That data is then converted to rough nominal values $B_{xn}^{(k)}$, $B_{yn}^{(k)}$, and $B_{zn}^{(k)}$ for $0 \leq k \leq k_0$ by some nominal calibration. Next, the latitude and longitude of each sample must be calculated to obtain a reference field $B_{x0}^{(k)}$,

$B_{y0}^{(k)}$, and $B_{z0}^{(k)}$. Finally, a reference magnetic angle α_k must be computed at each reference point where

$$\alpha_k = 4 \arctan \left(B_{z0}^k, \sqrt{(B_{x0}^{(k)})^2 + (B_{y0}^{(k)})^2} \right).$$

Once the preliminary steps have been performed, the algorithm to find the direction-dependent coefficients can be used. It is assumed that the magnitude-dependent coefficients have already been calculated. The method proceeds as described in Algorithm 3.3.1.

```
Initialize $\hat{\mathbf{x}}_0 = \left[ \begin{array}{cccccc} c_{12}^{(0)} & c_{13}^{(0)} & c_{21}^{(0)} & c_{23}^{(0)} & c_{31}^{(0)} & c_{32}^{0} \end{array} \right]^T$;
Initialize $P_0 = \mathrm{diag}([\begin{array}{cccccc} P_{12} & P_{13} & P_{21} & P_{23} & P_{31} & P_{32} \end{array}])$ and $R$ which are found empirically;
for $k = 1, \ldots, k_0$ do
    Form $C_2^{(k-1)} = \left[ \begin{array}{cccc} 1 & c_{12}^{(k-1)} & c_{13}^{(k-1)} & 0 \\ c_{21}^{(k-1)} & 1 & c_{23}^{(k-1)} & 0 \\ c_{31}^{(k-1)} & c_{32}^{(k-1)} & 1 & 0 \end{array} \right]$;
    $\mathbf{B}_2^{(k)} = C_2^{(k-1)} \mathbf{B}_1^{(k)}$ where $\mathbf{B}_1^{(k)}$ is nominal calibration data;
    $\beta^{(k)} = 4 \arctan(B_{z2}^{(k)}, \sqrt{(B_{x2}^{(k)})^2 + (B_{y2}^{(k)})^2})$;
    $\mathbf{g}_k^T = \left[ \begin{array}{cccccc} \frac{B_{x2}B_{y2}B_{z2}}{B_{t2}|B_2|^2} & \frac{B_{x2}B_{z1}B_{z2}}{B_{t2}|B_2|^2} & \frac{B_{x1}B_{y2}B_{z2}}{B_{t2}|B_2|^2} & \frac{B_{y2}B_{z1}B_{z2}}{B_{t2}|B_2|^2} & -\frac{B_{t2}B_{x1}}{|B_2|^2} & -\frac{B_{t2}B_{y1}}{|B_2|^2} \end{array} \right]$ where
    $B_{t2}^{(k)} = \sqrt{(B_{x2}^{(k)})^2 + (B_{y2}^{(k)})^2}$ and $\mathbf{g}_k$ is a linearized observation vector.;
    $K_k = \frac{P_{k-1}\mathbf{g}_k}{(\mathbf{g}_k^T P_{k-1} \mathbf{g}_k + w^2 R)}$ where $0 < w \leq 1$ is a forgetting factor.;
    $\hat{\mathbf{x}}_k = K_k(\alpha_k - \beta_k) + \hat{\mathbf{x}}_{k-1}$ where $\alpha_k$ is known before hand.;
    $P_k = (I - K_k g_k^T) P_{k-1} / w^2$;
end
```

Algorithm 3.3.1: RLS Magnetometer Calibration

3.4 Angular Rate Estimation

In [24] the authors propose an Extended Kalman Filter with time propagation based on the solution of Jacobian elliptic functions to estimate a satellite's angular rates from magnetometer measurements. No modeling of the earth's magnetic field is necessary. Further, no knowledge of the satellite's attitude is needed. The satellite is assumed to be in a tumbling mode, which means that the only applied torques are stochastic torques (i.e. there is no input).

The satellite's dynamics are modeled by

$$\dot{\omega} = J^{-1}(-\omega \times J\omega) + \xi$$

or

$$\dot{\mathbf{x}} = f(x, J) + \xi \tag{3.8}$$

where ω is a vector of angular rates, J is the inertia matrix, and $\xi \sim \mathcal{N}(0, Q_c)$ is Gaussian noise. The observation model is based on

$$\frac{d\breve{\mathbf{b}}}{dt} = \frac{\partial \breve{\mathbf{b}}}{\partial t} + \omega \times \breve{\mathbf{b}} \tag{3.9}$$

where $\breve{\mathbf{b}}$ is the earth magnetic field vector, $\frac{d\breve{\mathbf{b}}}{dt}$ is the derivative of the magnetic field vector in an inertial reference frame, and $\frac{\partial \breve{\mathbf{b}}}{\partial t}$ is the derivative of the magnetic field in the body reference frame. For most orbits $\frac{d\breve{\mathbf{b}}}{dt} \approx 0$ so

$$\frac{\partial \breve{\mathbf{b}}}{\partial t} \approx -\omega \times \breve{\mathbf{b}} = [\breve{b}\times]\omega$$

where

$$[\breve{b}\times] = \begin{bmatrix} 0 & -\breve{b}_x & \breve{b}_y \\ \breve{b}_z & 0 & -\breve{b}_x \\ -\breve{b}_y & \breve{b}_x & 0 \end{bmatrix}.$$

A continuous model must be discretized since sensor measurements are not continuous. The nonlinear state equation is discretized to

$$\mathbf{x}_{k+1} = \phi_k \mathbf{x}_k + \mathbf{u}_k$$

where ϕ_k is the linearized state transition matrix, and $\mathbf{u}_k \sim \mathcal{N}(0, Q)$ where $Q = Q_c \Delta t$. Instead of conventional integration methods at the time propagation step of the EKF, the authors propose using the analytical solution for rigid body motion in terms of the Jacobian elliptic function. They assert that this greatly reduces the number of FLOPs required in computation. The linearized state transition matrix is used to propagate the error covariance matrix P_k. The transition matrix is approximated as

$$\phi_k \approx I + F_k \Delta t$$

where

$$F_k = \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}} = \begin{bmatrix} 0 & \frac{(J_{yy}-J_{zz})\hat{x}_3}{J_{xx}} & \frac{(J_{yy}-J_{zz})\hat{x}_2}{J_{xx}} \\ \frac{(J_{zz}-J_{xx})\hat{x}_3}{J_{yy}} & 0 & \frac{(J_{zz}-J_{xx})\hat{x}_1}{J_{yy}} \\ \frac{(J_{xx}-J_{yy})\hat{x}_2}{J_{zz}} & \frac{(J_{xx}-J_{yy})\hat{x}_1}{J_{zz}} & 0 \end{bmatrix},$$

and J is assumed to be diagonal.

The magnetometer measurement model used is

$$\tilde{\mathbf{b}}_k = \breve{\mathbf{b}}_k + \nu_k$$

where $\nu_k \sim \mathcal{N}(0, R_{TAM})$ is the stationary measurement noise. The body-referenced temporal derivative is approximated with a backwards difference. Then the observation equation is

$$\mathbf{z}_k = H_k \mathbf{x}_k + \mathbf{n}_k$$

where $H_k = [\tilde{b}_{k-1}\times]\Delta t$, $\mathbf{z}_k = \tilde{\mathbf{b}}_k - \tilde{\mathbf{b}}_{k-1}$ is the effective measurement vector, and $\mathbf{n}_k = \nu_k - \nu_{k-1}$ is the effective measurement noise, which is colored.

The EKF can be augmented to deal with the colored noise. n_k can be modeled as a first-order Markov process

$$\mathbf{n}_k = \phi_k^c \mathbf{n}_{k-1} + \mathbf{w}_{k-1}$$

where $\mathbf{w}_k \sim \mathcal{N}(0, R_k^w)$ and ϕ_k^c is related to the process autocorrelation. This yields

$$\begin{aligned} \phi_k^c &= -0.5I = \phi^c \\ R_k^w &= 1.5R_{TAM} = R^w. \end{aligned}$$

The original EKF state vector is augmented with $\overline{\mathbf{x}}_k = \begin{bmatrix} \mathbf{x}_k^T & \mathbf{n}_k^T \end{bmatrix}^T$. The new state transition matrix, observation matrix, and Q matrix are

$$\begin{aligned} \overline{\phi}_k &= \begin{bmatrix} \phi_k & 0 \\ 0 & \phi^c \end{bmatrix} \\ \overline{H}_k &= \begin{bmatrix} H_k & I_{3\times 3} \end{bmatrix} \\ \overline{Q} &= \begin{bmatrix} Q & 0 \\ 0 & R^w \end{bmatrix}. \end{aligned}$$

The entries in Q are found through a trial-and-error tuning process. Augmenting the system results in a singular measurement model. This difficulty can be overcome by replacing the singular measurement noise by a small positive-definite matrix in the Kalman filter measurement update equation or implementing a reduced-order filter.

3.5 Conclusions

Several algorithms have been presented in this chapter to estimate the constant bias of a magnetometer. In order for an algorithm to be suitable for estimating the bias of a magnetometer on a rocket, the algorithm must be fast since all bias estimation must occur while the rocket is still in the launch mechanism. The scoring method has the benefit of being recursive but requires taking partial derivatives, which is time consuming. This method can also become stuck in local minima. The fixed point method also has the benefit of being iterative, but it requires taking the inverse of a matrix, a costly operation. This algorithm also exhibits poor convergence, which makes it unsuitable for our rocket problem. Davenport's method of estimating bias has the disadvantage of producing inconsistent bias estimates, and this algorithm requires taking the inverse of a matrix, which makes it too slow for our problem. The TWOSTEP algorithm requires a good starting point for estimates of bias, which is not available in the rocket problem so this method of bias estimation is unsuitable. The recursive least squares method has the advantage of being a quick and simple method to estimate biases. However, this algorithm is not suitable since the rocket does has neither the orientation required by the algorithm nor is it spinning slowly as required by the algorithm. Acuña's algorithm is the best algorithm for calibrating a magnetometer on a rocket in a launch mechanism. This algorithm does not require a model and is simple to implement. It does require taking the inverse of a matrix, but this computational cost can be minimized by using as few measurements as possible in constructing the matrix to be inverted.

We have also presented a Kalman filter based algorithm for angular rate estimation using only a magnetometer. This algorithm has the benefits of not requiring a magnetic field model of the earth and no knowledge of the attitude of the vehicle is needed. However, this algorithm assumes the vehicle is tumbling and no input torques are applied making it unsuitable for use on a rocket. The use of magnetometers primarily on satellites has resulted in many algorithms that are useful for orbiting bodies but few that can be used on ballistic projectiles.

Chapter 4

Estimation And Control

Many methods have been developed to control missiles. In this chapter we present some of these control methods with emphasis on reaction jet control. The various sensor suites assumed to be present on the missiles are also considered. In §4.1 we present several works that address control laws for various kinds of missiles including skid-to-turn, air-to-air, surface-to-air, and surface-to-surface. Next, in §4.2 we present control methods for missiles with reaction jets. First, we present a variable-structure (sliding-mode) control method. A presentation of a time-optimal control method for missiles with a single reaction jet follows. In §4.2 we present control methods for missiles with multiple reaction jets. In §4.2.4 we conclude the section with a presentation of a control method that uses projectile linear theory. We draw some conclusions in §4.3.

4.1 Current Missile Control Methods

In the literature to date, little if any work has been presented on controlling a projectile solely during the first few seconds of flight. Most work focuses on maneuverable projectiles such as skid-to-turn (STT) missiles [25], [26], projectiles with thrust vectoring [27], and projectiles with other forms of control [28], [29]. Moreover, the projectiles presented in the literature have various sensor suites that allow them to perform tasks such as target tracking to aid in guidance and navigation [30], [31], [32]. The projectiles in the literature that do have thrusters or reaction jets also have other control mechanisms [33], [34], [35], [36], [37], [38].

Various control methods have been investigated to increase the accuracy and performance of missiles (air-to-air, surface-to-air, surface-to-surface). One of the simplest control techniques is linear control. In [35] time-varying pole placement is used to develop a controller while [39] use linear-parameter-varying control. Due to the complex dynamics of missiles and the uncertainties in system models, nonlinear control techniques are often applied to the problem. Backstepping is a relatively simple nonlinear control technique [40]. Sliding-mode control is a common technique applied to missiles [33], [41], [42]. In [26] a combination of sliding-mode control and adaptive feedback linearization is used. Dynamic inversion [43], [44] or input/output linearization [25] are also possible methods of deriving a solution. H^{∞} control is commonly used to obtain an optimal solution [45], [46], [31]. The use of fuzzy logic and neural networks is also common [47], [48], [32], [36], [49].

In [50] Kuhn addresses the issue of navigation of gun-launched munitions; however, a variety of sensor readings are assumed to be available. Kuhn proposes an Extended Kalman Filter (EKF) to calibrate gyros in-flight and suggests that better results can be obtained by including accelerometer and GPS data to improve estimates of aerodynamic drag. Kuhn notes that the use of low-cost sensors can result in uncompensated sensor errors such as random walk and nonlinearity. In other words, the use of inexpensive gyros alone is unreliable for navigation. Kuhn proposes the use of a magnetometer to improve state estimates.

An estimator to predict vehicle attitude and gyro bias is presented in [51]. Gyros are modeled to have random walk bias and additive noise. Estimates of attitude and gyro bias are obtained with the proposed state observer, which uses quaternions to represent attitude estimations obtained from the vehicle's kinematic equations. The gyro bias estimates are shown to converge exponentially to a root mean square (RMS) bound. However, for the bias estimates to converge, both the noise driving the random walk bias and the additive noise must be bounded.

In [52] Gebre-Egziabher *et al.* present a gyro-free method of determining attitude using only low-cost accelerometers and magnetometers. The work is based on Wahba's problem, which shows that attitude can be determined from two noncolinear vectors that are known in one frame and measured in another frame [1] (see Chapter 2). In the proposed estimator, the accelerometers are used to measure gravity in the body frame, and the magnetometers are used to measure the Earth's magnetic field in the body frame. Using the solution to Wahba's problem, the rotation is found that puts the body frame into an Earth-fixed frame.

Although much literature exists for navigational solutions for maneuverable projectiles with various control mechanisms and various sensor suites, little has been written about projectiles controlled solely by reaction jets. An overview of the existing literature is presented in §4.2. Even less has been written about controlling such projectiles with only gyros and accelerometers. The relatively recent rise in popularity of MEMS technology has yielded little work in controlling a projectile using only MEMS inertial sensors.

4.2 Reaction Jet Control

While much work has been developed for controlling missiles through aerosurfaces, relatively little attention has been given to control of missiles with on/off reaction jets. In this section we first present an overview of work on variable-structure control, which has been developed for missiles with both aerosurfaces and reaction jets. We then present an overview of a method to control a missile with only a single reaction jet. We conclude the section with a presentation of a control method for the case of a missile with multiple reaction jets.

4.2.1 Variable-Structure Control

Variable-structure control, also called sliding-mode control, has the defining characteristic that system states are driven discontinuously toward a hyperplane in state space. The control is designed so that all motion near the hypersurface is directed toward the hypersurface. The states' approach to the hypersurface will be at least asymptotic. This is the type of control law developed for missiles that are controlled with both aerosurfaces and reaction jets in [29], [37], and [53].

In [53] Weil and Wise develop an autopilot based on variable-structure control for a missile at a trim flight condition. The nonlinear equations that describe the missile's dynamics are linearized about the trim conditions. The model has two sets of inputs: fin deflection and reaction jet condition. The linearized equations are used to derive a linear switching surface for the reaction jet control and a linear feedback controller for the fins. The problem is then split into two linear quadratic regulator (LQR) problems, which are solved to find the optimal balance between fin actuator rates and reaction jet output.

Thukral and Innocenti use variable-structure control systems to develop an autopilot so that a high-performance missile with both on/off reaction jets and control fins can perform a 180°

maneuver to reverse heading [29], [37]. This maneuver requires the missile to go through three phases of flight: pitching to the point of stall, turning while stalled, and recovery from the stall in the desired attitude and direction. Of particular interest is the second phase in which only the reaction jets control the missile's performance. In the first and third stages both the fins and reaction jets are used. Thukral and Innocenti develop a model for the missile's behavior during each phase of the maneuver, since the aerodynamic properties vary widely from stage to stage. In each model the missile is treated as a rigid body with a first bending-mode natural frequency of about 30 Hz [29]. Aerodynamic data for the modeled missile is not available for the second stage of the maneuver, so Thukral and Innocenti base their model on a worst-case scenario with attitude described by

$$I_y\dot{q} = QSC_N L_{cp} + L_{RCS}T_{RCS}u_T \tag{4.1}$$

where I_y is the pitch axis moment of inertia, $\dot{q}$ is pitch angular acceleration, Q is dynamic pressure, S is cross sectional area, C_N is the normal force coefficient, L_{cp} is the distance between the center of pressure and the center of mass, L_{RCS} is the jets' moment arm, T_{RCS} is the reaction jet thrust, and u_T is the throttle variable for the reaction jets.

Variable-structure control is applied to the second-stage dynamics by choosing a desired model for the missile's pitch rotational dynamics from which error dynamics are defined. The desired model has the form

$$\dot{\overline{\mathbf{x}}}_m = A_m\overline{\mathbf{x}}_m + B_m\mathbf{u}_m, \tag{4.2}$$

and the actual system dynamics are

$$\dot{\overline{\mathbf{x}}} = A\overline{\mathbf{x}} + B\mathbf{u}_\mathbf{T}. \tag{4.3}$$

The error dynamics are modeled by

$$\dot{\overline{\mathbf{e}}} = A_m\overline{\mathbf{e}} + [A_m - A]\overline{\mathbf{x}} + B_m\mathbf{u}_m - D - B\mathbf{u}_T \tag{4.4}$$

where D is a disturbance vector, $\overline{\mathbf{e}} = \overline{\mathbf{x}}_m - \overline{\mathbf{x}}$. The control law implementation (nonlinear) is given by

$$\mathbf{u}_T = \text{sgn}(s) \tag{4.5}$$

$$s = G\overline{\mathbf{e}} \tag{4.6}$$

where G is chosen to ensure a desirable response during sliding [29].

4.2.2 Time-Optimal Control with A Single Reaction Jet

In [54], [55], [56], and [57] Jahangir and Howe develop methods to control the attitude of a missile with a single thruster located at right angles to the missile's spin axis. The final desired attitude is known, and the goal is to drive the missile to the desired attitude in the minimum amount of time. The missile, which is assumed to have a large roll rate, is controlled by turning on the thruster for a short period of time during each roll cycle. In [56] Jahangir and Howe develop a method to generate thruster firing times that does not require the solution of a Two-Point Boundary-Value Problem (TPBVP). In [54] and [55] Jahangir and Howe develop a method of control for the case when two pulses are sufficient to drive the missiles's attitude to the desired state. The remainder of this section summarizes [57], in which the Jahangir and Howe develop a method of attitude control for the case where two thruster pulses are insufficient to achieve the desired final attitude.

Jahangir and Howe develop a state vector of dimensionless variables and then transform this vector so that boundary-condition points with known thruster firing times can be generated without the need to solve a TPBVP. To this end the following variables are defined

$$\begin{aligned}
\Omega_y &= \frac{\omega_y}{\omega_x} && (4.7)\\
\Omega_z &= \frac{\omega_z}{\omega_x} && (4.8)\\
\lambda_y &= \frac{M_y}{I_y \omega_x^2} && (4.9)\\
A &= 1 - \frac{I_x}{I_y} = 1 - \frac{I_x}{I_z} && (4.10)\\
T &= \omega_x t && (4.11)
\end{aligned}$$

where ω_x, ω_y, and ω_z are rotational rates, I_x, I_y, and I_z are the missile's moments of inertia, M_y is the moment applied about the missile's y-axis in the body frame, and t is time. Then the state equations are given by

$$\begin{aligned}
\dot{\Omega}_y &= A\Omega_z + \lambda_y && (4.12)\\
\dot{\Omega}_z &= -A\Omega_y && (4.13)\\
\dot{\psi} &= (\Omega_y \sin(\phi) + \Omega_z \cos(\phi)) \sec(\theta) && (4.14)\\
\dot{\theta} &= \Omega_y \cos(\phi) - \Omega_z \sin(\phi) && (4.15)\\
\dot{\phi} &= 1 + (\Omega_y \sin(\phi) + \Omega_z \cos(\phi)) \tan(\theta) && (4.16)
\end{aligned}$$

where all derivatives are taken with respect to T. Then the state space representation is given by

$$\begin{aligned}
\dot{\mathbf{x}} &= \begin{bmatrix} Ax_2 \\ -Ax_1 \\ (x_1 \sin(x_5) + x_2 \cos(x_5)) \sec(x_4) \\ x_1 \cos(x_5) - x_2 \sin(x_5) \\ 1 + (x_1 \sin(x_5) + x_2 \cos(x_5)) \tan(x_4) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \lambda_y && (4.17)\\
\dot{\mathbf{x}} &= f(\mathbf{x}) + \mathbf{g}u && (4.18)
\end{aligned}$$

where $\mathbf{x} = \begin{bmatrix} \Omega_y & \Omega_z & \psi & \theta & \phi \end{bmatrix}^T$ and the control input is $u = \lambda_y$. The initial state vector is

$$\mathbf{x}_0 = \begin{bmatrix} x_{1,0} & x_{2,0} & 0 & 0 & 0 \end{bmatrix}^T, \tag{4.19}$$

and the desired final state vector is

$$\begin{bmatrix} 0 & 0 & x_{3,d} & x_{4,d} & \text{free} \end{bmatrix}^T \tag{4.20}$$

where the fifth term can have any value since roll angle does not matter. The control input has a maximum u_{max} so the bounds on the control are given by

$$0 \leq u \leq u_{max}.$$

The solution of the developed system involves a TPBVP (an iterative problem) so thruster firing times for the desired boundary conditions are found off-line and stored in a look-up table. The control is real-time implementable by table look-up and interpolation. Another approach is to use a transformation that allows the boundary condition points for which thruster firing times are known to be generated without the requirement of an iterative solution of a TPBVP. By using a transformation of the state vector, time-optimal trajectories can be computed by integrating backwards from the desired final condition.

The transformation begins by defining a new reference frame. For this new frame, the axes are fixed in the target with the x-axis pointing in the desired direction of the missile's body frame x-axis. The Euler angles that convert from the missile's body frame to the new frame are y_3, y_4, and y_5 corresponding to yaw, pitch, and roll, respectively. Define $y_1 = x_1$ and $y_2 = x_2$. Then the new state vector is

$$\mathbf{y} = \begin{bmatrix} y_1 & y_2 & y_3 & y_4 & y_5 \end{bmatrix}^T, \tag{4.21}$$

and the transformed system is

$$\dot{\mathbf{y}} = f(\mathbf{y}) + \mathbf{g}u. \tag{4.22}$$

The initial and final conditions are given by

$$\begin{aligned}\mathbf{y}(T_0) &= \mathbf{y}_0 = \begin{bmatrix} y_{1,0} & y_{2,0} & y_{3,0} & y_{4,0} & y_{5,0} \end{bmatrix}^T & (4.23)\\ \mathbf{y}(T_f) &= \mathbf{y}_f = \begin{bmatrix} 0 & 0 & 0 & 0 & \text{free} \end{bmatrix}^T, & (4.24)\end{aligned}$$

respectively. The transformation from the new frame to the body frame is given by

$$C(x_{3,d}, x_{4,d}, x_{5,d}) = \Phi C(y_3, y_4, y_5)^T \qquad (4.25)$$

where

$$\begin{aligned}C(\psi,\theta,\phi) &= \begin{bmatrix} c_\psi c_\theta & s_\psi c_\theta & -s_\theta \\ -s_\psi c_\phi + c_\psi s_\theta s_\phi & c_\psi c_\phi + s_\psi s_\theta s_\phi & c_\theta s_\phi \\ s_\psi s_\phi + c_\psi s_\theta c_\psi & -c_\psi s_\phi + s_\psi s_\theta c_\phi & c_\theta c_\phi \end{bmatrix} & (4.26)\\ \Phi &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(y_{5,f}) & \sin(y_{5,f}) \\ 0 & -\sin(y_{5,f}) & \cos(y_{5,f}) \end{bmatrix}. & (4.27)\end{aligned}$$

c_i and s_i are $\cos(i)$ and $\sin(i)$, respectively.

With the transformation defined, a time-optimal control law can be found. Define the cost function as

$$J = \int_{T_0}^{T_f} 1dt. \qquad (4.28)$$

Necessary conditions for optimality are obtained from the Hamiltonian

$$H = \mathbf{q}^T \dot{\mathbf{y}} - 1 \qquad (4.29)$$

where $\mathbf{q}$ is the costate vector. This leads to an optimal control u^* from the conditions

$$\begin{aligned}
\dot{\mathbf{y}}^* &= \frac{\partial H}{\partial \mathbf{q}} = f(\mathbf{y}^*) + \mathbf{g}u^* && (4.30)\\
\dot{\mathbf{q}} &= -\frac{\partial H}{\partial \mathbf{y}} = H(\mathbf{y}^*)\mathbf{q}^* && (4.31)\\
u^* &= \begin{cases} u_{max} & \text{if } q_1^* > 0 \\ 0 & \text{if } q_q^* < 0 \end{cases} && (4.32)
\end{aligned}$$

where $H(\mathbf{y}) = -\partial f / \partial \mathbf{y}$. The boundary conditions of the Hamiltonian and costate vector are

$$\begin{aligned}
H(T_f) &= 0 && (4.33)\\
\mathbf{q}(T_0) &= \mathbf{q}_0 = \begin{bmatrix} \text{free} & \text{free} & \text{free} & \text{free} & \text{free} \end{bmatrix}^T && (4.34)\\
\mathbf{q}(T_f) &= \mathbf{q}_f = \begin{bmatrix} \text{free} & \text{free} & \text{free} & \text{free} & 0 \end{bmatrix}^T . && (4.35)
\end{aligned}$$

A control history can be generated with the following steps:

1. Initialize $\mathbf{y}_f$ and $\mathbf{q}_f$.

2. Integrate the $\dot{\mathbf{y}}$ and $\dot{\mathbf{q}}$ equations backward in time, and at each $T'_n = n\Delta T$ obtain $\mathbf{y}(T'_n)$ where ΔT is the time interval chosen to give desired data point spacing between $\mathbf{y}(T'_n)$ and $\mathbf{y}(T'_{n+1})$ and n is a positive integer.

3. Transform $\mathbf{y}(T'_n)$ to obtain the boundary conditions in the original form $x_{1,0}$, $x_{2,0}$, $x_{3,d}$, and $x_{4,d}$. Store the thruster switching times as functions of these four variables.

4.2.3 Multiple-Reaction Jet Control

Costello *et al.* [58], [59], [60] are the only authors that investigate control of a projectile solely through multiple reaction jets. In [58] a rocket is simulated that has a main thruster that has a limited burn time and is stabilized by three pop-out fins. Lateral pulse jets, each of which can only be fired once, are located toward the front of the rocket. Lateral pulse jet control is investigated to improve the dispersion pattern of direct-fire atmospheric rockets. This method of control falls under the category of impulse control, unlike other common rocket control methods such as proportional navigation guidance (PNG), which is continuous. In [58] Jitpraphai *et al.* investigate PNG, parabolic and proportional navigation guidance (PAPNG), and trajectory tracking (TT) applied to a direct-fire atmospheric rocket equipped with a lateral pulse jet control mechanism. All three control laws use the same control logic to fire the lateral pulse jets.

Proportional Navigation Guidance (PNG)

The guidance law for PNG is

$$A_C = NV_c\dot{\lambda} \tag{4.36}$$

where N is the proportional navigation constant (typically valued between 3 and 5), V_C is rocket closing velocity, and $\dot{\lambda}$ is line-of-sight (LOS) angular rate. PNG has horizontal plane and vertical plane control law components and three reference frames are used: the LOS frame, the target frame, and the inertial frame. For the following equations, the target is assumed to be stationary. All axes in [58] are labeled (I, J, K) with a subscript to indicate the reference frame. The I-axis of the LOS frame points from the rocket to the target.

The horizontal component of λ is then

$$\lambda_H = \tan^{-1}\left(\frac{y_T - y}{x_T - x}\right) \tag{4.37}$$

and

$$\dot{\lambda}_H = \frac{-\dot{y}(x_T - x) + \dot{x}(y_T - y)}{(x_T - x)^2 + (y_T - y)^2}. \tag{4.38}$$

Then the horizontal acceleration command is

$$\begin{aligned} \tilde{A}_{YC} &= N_H\dot{\lambda}_H u_H && (4.39) \\ u_H &= \cos(\lambda_H)\dot{x} + \sin(\lambda_H)\dot{y} && (4.40) \end{aligned}$$

where the $\sim$ indicates the LOS frame. The vertical plane component is given as

$$\begin{aligned}
\lambda_V &= \tan^{-1}\left(\frac{z_T - z}{x_T - x}\right) && (4.41)\\
\dot{\lambda}_V &= \frac{-\dot{z}(x_T - x) + \dot{x}(z_T - z)}{(x_T - x)^2 + (z_T - z)^2} && (4.42)\\
\tilde{A}_{ZC} &= N_V \dot{\lambda}_V u_V && (4.43)\\
u_V &= \cos(\lambda_V)\dot{x} + \sin(\lambda_Y)\dot{z} && (4.44)
\end{aligned}$$

The total acceleration command is then

$$\tilde{A}_C = \tilde{A}_{YC} J_L + \tilde{A}_{ZC} K_L = A_{XC} I_B + A_{YC} J_B + A_{ZC} K_B \qquad (4.45)$$

where (I, J, K) are unit vectors. The input to the lateral pulse ring firing logic is the magnitude of the off-axis command acceleration. This magnitude is given by

$$\Gamma = \sqrt{A_{YC}^2 + A_{ZC}^2} \qquad (4.46)$$

and, the error signal phase is given by

$$\gamma = \tan^{-1}(A_{ZC}/A_{YC}). \qquad (4.47)$$

Parabolic and Proportional Navigation Guidance (PAPNG)

PAPNG uses the same horizontal control law as PNG. In the vertical plane, a desired parabolic trajectory is described as

$$\hat{z}_P = \hat{z}_T + K_1 \hat{x}_p + K_2 \hat{x}_P^2 \qquad (4.48)$$

where the $\hat{x}_P$ and $\hat{z}_P$ are components of the rocket position in the target frame, $\hat{x}_T$ and $\hat{z}_T$ are components of the target position in the target frame, and K_1 and K_2 are defined by Equations (4.49)-(4.50). The desired angle at which the rocket passes through the target is defined as β_f.

Then

$$
\begin{aligned}
K_1 &= \tan(\beta_F) && (4.49)\\
K_2 &= \frac{-(K_1\sqrt{(x_T-x)^2+(y_T-y)^2}+\hat{z}_T-\hat{z}_P)}{(x_T-x)^2+(y_T-y)^2}. && (4.50)
\end{aligned}
$$

The command acceleration in the vertical LOS plane is

$$
\tilde{A}_{ZC} = \frac{V_L(\beta_D-\beta)}{\tau} \qquad (4.51)
$$

where V_L is the rocket velocity magnitude in the LOS frame, β is the flight path angle in the LOS frame, τ is the acceleration command time constant, and β_D is the rocket's desired flight path angle. The acceleration command is then given by

$$
\begin{aligned}
\tilde{A}_C &= -\tilde{A}_{ZC}\sin(\beta)I_L + \tilde{A}_{YC}J_L - \tilde{A}_{ZC}\cos(\beta)K_L && (4.52)\\
&= A_{XC}I_B + A_{YC}J_B + A_{ZC}K_B. && (4.53)
\end{aligned}
$$

Trajectory Tracking (TT)

In this method a command trajectory is assumed to be known prior to launch. From this information the error between the desired path and the actual path can be computed. The magnitude of this error is used as control input to the thruster firing logic. The magnitude of the error and the error phase angle are

$$
\begin{aligned}
\Gamma &= \sqrt{e_Y^2+e_Z^2} && (4.54)\\
\gamma &= \tan^{-1}(e_z/e_Y). && (4.55)
\end{aligned}
$$

Lateral Pulse Jet Firing Logic

In order for a jet to fire, four requirements must be met. The following two requirements apply to all lateral jets.

- The error magnitude must be greater than some tolerance.
- The time between thruster firings must be greater than some threshold.

The next two requirements are for a specific lateral jet.

- The difference between the error phase angle and the individual pulse jet force must be less than some threshold angle.
- The pulse jet has not been previously fired.

The last condition enforces the requirement that each jet can only be fired once.

In [59] the path control system is designed to track a specific trajectory. To do this the rocket's measured position is compared to the commanded position to create an error signal, which in the rocket body frame is given by

$$\begin{bmatrix} e_X \\ e_Y \\ e_Z \end{bmatrix} = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\phi + c_\phi c_\psi & s_\phi c_\theta \\ c_\phi s_\theta c_\phi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} x_C - x \\ y_C - y \\ z_C - z \end{bmatrix} \tag{4.56}$$

where ϕ, θ, and ψ are the roll, pitch, and yaw Euler angles. Magnitude and phase error are defined as

$$\Gamma = \sqrt{e_Y^2 + e_Z^2} \tag{4.57}$$

$$\gamma = \tan^{-1}(e_z/e_Y). \tag{4.58}$$

With this information available at each computation cycle, four criteria are checked to see if an individual thruster should be fired (lateral jet thrust is modeled as a constant of a known duration). The four conditions are

1. The tracking error magnitude must be greater than some threshold

$$\Gamma > e_{thres}. \tag{4.59}$$

2. A certain amount of time must have passed since the last jet firing

$$t - t^* > \Delta t_{thres} \tag{4.60}$$

where t^* is the time of the most recent pulse jet firing.

3. The angle between the tracking error and the individual pulse jet under consideration must be less than a threshold

$$|\theta_i - pi - \gamma - \dot{\gamma}(\Delta_{PJ}/2)| < \delta_{thres} \tag{4.61}$$

where θ_i is the angle between the body frame y-axis and the ith pulse jet and Δ_{PJ} is the pulse jet firing duration.

4. The pulse jet under consideration must not have been previously fired.

The purpose of Δt_{thres} is to ensure that the rocket has enough time to respond to a command to a particular pulse firing as well as to ensure that many pulse firings at once do not overcompensate for trajectory errors.

4.2.4 Projectile Linear Theory

In [60] Costello *et al.* utilize projectile linear theory to develop a control law. Dynamic equations have been developed to allow the closed-form solution of a missile under restricted flight conditions. The equations and the solutions to the equations have become known as projectile linear theory. Projectile linear theory uses a reference frame that is attached to the projectile but does not roll. A dimensionless variable arc length s is used rather than time, and a change of variables from the velocity along the projectile's axis of symmetry to total velocity V is used. The model also assumes that Euler pitch and yaw angles and the aerodynamic angle of attack are small.

Costello *et. al.* convert their model of a projectile according to this theory to yield

$$
\begin{aligned}
\frac{y(s)}{D} &= \frac{y_0}{D} + (\psi_0 + v_0/V_0)s + \Big(\frac{\rho SD}{2m}\Big)\frac{(C_{X0} - C_{NA})}{V_0}\Big\{\frac{\Omega_{vf}}{\phi_f^2}[\exp(\lambda_f^* s)\sin(\phi_f s + \theta_{vf} - \pi) \\
&- \sin(\theta_{vf} - \pi) - \phi_f \cos(\theta_{vf} - \pi)s]\Big\} \\
&+ \Big(\frac{\rho SD}{2m}\Big)\frac{(C_{X0} - C_{NA})}{V_0}\Big\{\frac{\Omega_{vs}}{\phi_s^2}[\exp(\lambda_s^* s)\sin(\phi_s s + \theta_{vs} - \pi) \\
&- \sin(\theta_{vs} - \pi) - \phi_s \cos(\theta_{vs} - \pi)s]\Big\} \qquad (4.62) \\
\frac{z(s)}{D} &= \frac{z_0}{D} + (-\theta_0 + w_0/V_0)s + gD\Big[\frac{m}{\rho SDV_0C_{X0}}\Big]^2 \times \Big\{\exp\Big(\frac{\rho SDC_{X0}s}{m}\Big) \\
&- \frac{\rho SDC_{X0}s}{m} - 1\Big\}\Big(\frac{\rho SD}{2m}\Big)\frac{C_{X0} - C_{NA}}{V_0}\Big\{\frac{\Omega_{wf}}{\phi_f^2}[\exp(\lambda_f^* s)\sin(\phi_f s + \theta_{wf} - \pi) \\
&- \sin(\theta_{wf} - \pi) - \phi_f \cos(\theta_{wf} - \pi)s]\Big\} \\
&+ \Big(\frac{\rho SD}{2m}\Big)\frac{(C_{X0} - C_{NA})}{V_0}\Big\{\frac{\Omega_{ws}}{\phi_s^2}[\exp(\lambda_s^* s)\sin(\phi_s s + \theta_{ws} - \pi) - \sin(\theta_{ws} - \pi) \\
&- \phi_s \cos(\theta_{ws} - \pi)s]\Big\}. \qquad (4.63)
\end{aligned}
$$

D is projectile characteristic length, (u, v, w) are the translational velocity components of the projectile center of mass resolved in the fixed plane reference frame, V is the magnitude of the mass center velocity, ρ is air density, m is projectile mass, C_{NA} and C_{X0} are aerodynamic coefficients, and (x, y, z) are position vector components of the projectile mass center expressed in the inertial reference frame. Ω_{vf}, Ω_{vs}, λ_f^*, λ_s^*, ϕ_f, ϕ_s, θ_{vf}, θ_{vs}, Ω_{wf}, Ω_{ws}, θ_{wf}, and θ_{ws} are all constants. These equations are used to predict the impact point of the projectile given the current state of the system which is obtained from an IMU.

Thus, the problem is recast to control the impact point of the projectile in the target plane rather than controlling the trajectory. Control input is based on the difference between the actual target location and the predicted impact point. Costello *et. al.* make two key assumptions: the control law has full state feedback (x, y, z, ψ, θ, ϕ, u, v, w, p, q, and r) and the projectile has been provided with the target's inertial coordinates. The distance by which the projectile is predicted to miss the target is calculated as

$$\epsilon = \sqrt{(y_T - y_I)^2 + (z_T - z_I)^2} \qquad (4.64)$$

where the subscript T indicates the target frame and the subscript I indicates the inertial frame. The miss distance phase angle is defined as

$$\gamma = -\tan^{-1}\left(\frac{y_t - y_I}{z_T - z_I}\right) \tag{4.65}$$

A radius S_W defines a circle around the target in which it is desired for the missile to impact. As the missile gets closer to the target, S_W gets smaller. If the computed miss distance is greater than S_W, a thruster is fired. Linear impact theory is used to determine if firing a specific thruster will result in the missile hitting the target area.

Simple logic is used to decide if a pulse jet should be fired. A line segment from the target to the predicted impact point is computed via

$$S_p = (y_I - y_T)/(z_I - z_T). \tag{4.66}$$

Then if the two inequalities

$$\begin{aligned} y_I &\geq [(y_T - 1)/S_p](z_I - z_T) \qquad (4.67)\\ y_I^* &\geq [(y_T - 1)/S_p](z_I^* - z_T) \qquad (4.68) \end{aligned}$$

are both true or both false (meaning both points are on the same side of a curve that defines an allowable overshoot region) a jet is fired. The asterisk denotes controlled impact point coordinates. Next, ϵ is calculated. If ϵ is smaller than the allowable overshoot, a thruster is fired regardless of the results of the two inequalities.

4.3 Conclusions

We have presented several control methods in this chapter. Those methods that apply to missiles with control actuators other than reaction jets cannot be applied to our problem since the only control method available is reaction jets. Innocenti and Thukral present a sliding mode control law for a missile during a flight phase in which only reaction jets are available. This flight phase is well after the missile has been launched; therefore, the missile is not subjected to the disturbances of initial flight while only reaction jet control is used. The sliding mode control law presented does not address the challenges of flight immediately after launch to which our rocket is subjected. Further, sliding mode control is subject to chattering, oscillation about the sliding surface. Chattering is not acceptable in a problem in which a rocket must be controlled accurately within seconds during large disturbances. Jahangir and Howe's control method is for a rocket controlled solely with a reaction jet, but their rocket has a single reaction jet that can be fired multiple times. This method will not work for the rocket in this work because the rocket of concern has multiple, single-fire reaction jets. Jahangir and Howe's method also optimizes control of the rocket over the entire flight whereas the rocket of concern can only be controlled for a few seconds. Costello develops a few algorithms for a rocket controlled with multiple, one-shot reaction jets. The PNG algorithm controls attitude in the horizontal and vertical planes, but requires multiple reference frames including a target reference frame. For our problem a target reference frame is unavailable. The PAPNG method uses a desired parabolic path in the vertical frame to derive control commands, but this method also requires the use of a target reference frame. The TT method uses the error between a desired path and the actual path of the rocket. This method is best suited for controlling the rocket of this work since the TT method assumes multiple, one-shot reaction jets and needs no information about the target, only a desired trajectory to the target. This method can be used for the first few seconds of flight as well as over the entire flight. The control law based on linear projectile theory proposed by Costello is not suitable for our problem since this method assumes full state feedback, which is unavailable to the rocket in this work.

Chapter 5

Rocket Dynamics and State Estimation

In this chapter the dynamics for a ballistic projectile are developed as well as methods for estimating the projectile's attitude. As shown in Figure 2.1, the body frame axes are located at the rocket's center of gravity with the x-axis passing through the nose. The rocket is symmetric about the x-axis, which means that the moment of inertia in the y-direction is equal to the moment of inertia in the z-direction. The rocket has a main thruster that propels it and a ring of lateral thrusters located at the back of the rocket to provide control. Fins, which pop open shortly after the rocket clears the launch mechanism, are also located at the rocket's rear. As for the notation in this chapter, all zero-mean white noise values given by the variable η are independent. All constant biases are assumed to be estimated before the rocket leaves the launch platform; therefore, they do not appear in the estimation schemes.

5.1 Rotational Rate Modeling

The rotational dynamics of a rigid body are given by

$$\frac{d\omega}{dt} = J^{-1}(\tau - \omega \times (J\omega)) \tag{5.1}$$

where $\omega = \begin{bmatrix} p & q & r \end{bmatrix}^T$ is a vector of rotational rates, J is the moment of inertia matrix, and τ is a vector of torques (moments) applied to the body frame. For a rotating rocket, the applied moments are

$$\tau = \tau_{aero} + \tau_{prop} + \tau_{control}$$

where τ_{aero}, τ_{prop}, and $\tau_{control}$ respectively refer to the moments introduced by the aerodynamic, propulsion, and attitude control systems. Similarly, the forces that affect the rocket's body frame are given by

$$\mathbf{F} = \mathbf{F}_{grav} + \mathbf{F}_{aero} + \mathbf{F}_{prop} + \mathbf{F}_{control}$$

where the subscripts *grav*, *aero*, *prop*, and *control* refer to the gravitational, aerodynamic, propulsion, and control forces, respectively.

Wind affects the rocket's angle of attack α_{tot}. In the body frame x-direction, the wind is assumed to be zero. Angle of attack is computed from the relative velocity

$$\mathbf{v}_r = \begin{bmatrix} v_{r,x} \\ v_{r,y} \\ v_{r,z} \end{bmatrix}$$

as

$$\tan \alpha_{tot} = \frac{\left\| \begin{bmatrix} v_{r,y} \\ v_{r,z} \end{bmatrix} \right\|_2}{v_{r,x}}.$$

The wind azimuth angle is then

$$\tan \phi_b = \frac{v_{r,y}}{v_{r,x}}.$$

The aerodynamic moments are given by

$$\tau_{aero} = \overline{q} S_{ref} D_{ref} \left(r_{ref} \begin{bmatrix} c_{Lp}(t) & 0 & 0 \\ 0 & c_{mqr}(t) & 0 \\ 0 & 0 & c_{mqr}(t) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} c_L(t) \\ c_M(t) \\ c_N(t) \end{bmatrix} \right) \tag{5.2}$$

where $\overline{q}$ is dynamic pressure, S_{ref}, D_{ref}, and r_{ref} are constants, $c_{Lp}(t)$ and $c_{mqr}(t)$ are dynamic derivatives that are a function of Mach number, and $c_L(t)$, $c_M(t)$, and $c_N(t)$ are base coefficients. The roll base coefficient $c_L(t)$ is given by

$$c_L(t) = \phi_{fin} c_{LD}(t) + c_{L0}(t)$$

where ϕ_{fin} is the fin cant. $c_{LD}(t)$ and $c_{L0}(t)$ are functions of mach number when the fins are deployed and equal to zero otherwise. Then $c_L(t)$ can be written as

$$c_L(t) = e_f(\phi_{fin} c_{LD}(t) + c_{L0}(t))$$

where

$$e_f = \begin{cases} 0 & \text{if fins are closed} \\ 1 & \text{if fins are open} \end{cases}.$$

The pitch and yaw base coefficients are given by

$$\begin{bmatrix} c_M(t) \\ c_N(t) \end{bmatrix} = \begin{bmatrix} \cos(\phi_b) \\ -\sin(\phi_b) \end{bmatrix} c'_m(t)$$

where

$$c'_M(t) = c^w_M(t) + c^w_z(t) cg_x.$$

$c^w_M(t)$ is the wind frame pitch moment, $c^w_z(t)$ is the z-axis force coefficient, and cg_x is the x-coordinate center of gravity. Both $c^w_M(t)$ and $c^w_z(t)$ depend on angle of attack. Thus,

$$\tau_{aero} = \overline{q} S_{ref} D_{ref} \left(r_{ref} \begin{bmatrix} c_{Lp}(t)p \\ c_{mqr}(t)q \\ c_{mqr}(t)r \end{bmatrix} + \begin{bmatrix} e_f(\phi_{fin} c_{LD}(t) + c_{L0}(t)) \\ \cos(\phi_b) c'_m(t) \\ -\sin(\phi_b) c'_m(t) \end{bmatrix} \right).$$

The propulsion torque is given by

$$\tau_{prop} = \mathbf{r}_{prop} \times \mathbf{f}_{prop} + \begin{bmatrix} \tau_{motor} \\ 0 \\ 0 \end{bmatrix}$$

where

$$\mathbf{r}_{prop} = \begin{bmatrix} r_x \\ 0 \\ 0 \end{bmatrix} D_{ref}$$

is the motor lever arm with r_x being a constant. The propulsion force is given by

$$\mathbf{f}_{prop} = T(t) \begin{bmatrix} \cos(e_\psi)\cos(e_\theta) \\ \sin(e_\psi)\cos(e_\theta) \\ -\sin(e_\theta) \end{bmatrix}$$

where $T(t)$ is the main motor thrust and e_ψ and e_θ are motor misalignment angles. When τ_{motor} is assumed to be zero,

$$\tau_{prop} = \begin{bmatrix} 0 \\ r_x D_{ref}\sin(e_\theta)T(t) \\ r_x D_{ref}\sin(e_\psi)\cos(e_\theta)T(t) \end{bmatrix}. \tag{5.3}$$

Then from Equation (5.1), the rocket's rotational dynamics are

$$\begin{aligned} \dot{\omega} \;\; = & \begin{bmatrix} \frac{1}{I_{xx}(t)} & 0 & 0 \\ 0 & \frac{1}{I_{yy}(t)} & 0 \\ 0 & 0 & \frac{1}{I_{yy}(t)} \end{bmatrix} \left(\tau_{control} + \overline{q} S_{ref} D_{ref} \left(r_{ref} \begin{bmatrix} c_{Lp}(t) & 0 & 0 \\ 0 & c_{mqr}(t) & 0 \\ 0 & 0 & c_{mqr}(t) \end{bmatrix} \right.\right. \\ & \left.\begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} e_f(\phi_{fin} c_{LD}(t) + c_{L0}(t)) \\ cos(\phi_b) c'_m(t) \\ -\sin(\phi_b) c'_m(t) \end{bmatrix} \right) + \begin{bmatrix} 0 \\ r_x D_{ref}\sin(e_\theta)T(t) \\ r_x D_{ref}\sin(e_\psi)\cos(e_\theta)T(t) \end{bmatrix} - \\ & \left. \begin{bmatrix} 0 \\ (I_{xx}(t) - I_{yy}(t))pr \\ -(I_{xx}(t) - I_{yy}(t))pq \end{bmatrix} \right). \end{aligned}$$

The model then becomes

$$\dot{\omega} = \begin{bmatrix} \alpha_1(t) \\ \alpha_2(t) \\ \alpha_3(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{I_{xx}(t)} & 0 & 0 \\ 0 & \frac{1}{I_{yy}(t)} & 0 \\ 0 & 0 & \frac{1}{I_{yy}(t)} \end{bmatrix} \tau_{control}. \tag{5.4}$$

where

$$\begin{aligned} \alpha_1(t) &= (\overline{q}S_{ref}D_{ref}r_{ref}c_{Lp}(t)p + \overline{q}S_{ref}D_{ref}e_f(\phi_{fin}c_{LD}(t) + \\ &\quad c_{L0}(t)))/I_{xx}(t) \\ \alpha_2(t) &= (\overline{q}S_{ref}D_{ref}(r_{ref}c_{mqr}(t)q + cos(\phi_b)c'_m) + r_xD_{ref}\sin(e_\theta)T(t) - \\ &\quad (I_{xx}(t) - I_{yy}(t))pr)/I_{yy}(t) \\ \alpha_3(t) &= (\overline{q}S_{ref}D_{ref}(r_{ref}c_{mqr}(t)r - \sin(\phi_b)c'_m) + r_xD_{ref}\sin(e_\psi)\cos(e_\theta)T(t) + \\ &\quad (I_{xx}(t) - I_{yy}(t))pq)/I_{yy}(t) \end{aligned}$$

We make the following assumptions for the purposes of state estimation. First, due to the available sensors used for this problem, the angle of attack α cannot be determined. It is reasonable to assume that the angle of attack is small, so it is assumed to be zero. Second, the motor is assumed to be aligned properly so that e_ψ and e_θ are both zero. When the fins are closed, $e_f = 0$, and the model becomes

$$\begin{aligned} \dot{\omega} &= \begin{bmatrix} \overline{q}S_{ref}D_{ref}r_{ref}c_{Lp}(t)p/I_{xx}(t) \\ (\overline{q}S_{ref}D_{ref}r_{ref}c_{mqr}(t)q - (I_{xx}(t) - I_{yy}(t))pr)/I_{yy}(t) \\ (\overline{q}S_{ref}D_{ref}r_{ref}c_{mqr}(t)r + (I_{xx}(t) - I_{yy}(t))pq)/I_{yy}(t) \end{bmatrix} + \\ &\quad \begin{bmatrix} \frac{1}{I_{xx}(t)} & 0 & 0 \\ 0 & \frac{1}{I_{yy}(t)} & 0 \\ 0 & 0 & \frac{1}{I_{yy}(t)} \end{bmatrix} \tau_{control}. \end{aligned} \tag{5.5}$$

5.2 Pitch Rate and Yaw Rate Estimation

In this section the estimation of angular rates using only gyroscopes is considered. The output from the gyros is modeled as

$$\omega_s = \omega + w_i + b_i + \eta_i \tag{5.6}$$

where ω_s is the sensed angular rate, ω is the actual angular rate, w_i is a walking bias, b_i is a constant bias, and $i = \{x, y, z\}$. The walking bias dynamics are modeled as a first-order Markov process

$$\dot{w}_i = c_i w_i + \eta_i$$

where c_i is a constant and η_i is zero-mean Gaussian noise. Define τ as the time constant of the Markov process. Then $c_i = -\frac{1}{\tau}$. With the assumption that the roll rate is a known function, the system model from Equation (5.5) becomes a linear time-varying (LTV) system given by

$$\dot{\mathbf{x}} = \begin{bmatrix} a_{11}(t) & a_{12}(t) & 0 & 0 \\ -a_{12}(t) & a_{22}(t) & 0 & 0 \\ 0 & 0 & c_y & 0 \\ 0 & 0 & 0 & c_z \end{bmatrix} \mathbf{x} + \begin{bmatrix} \frac{1}{I_{yy}}(t) & 0 \\ 0 & \frac{1}{I_{yy}}(t) \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tau_{control}(t) + \mathbf{w}(t) \tag{5.7}$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \mathbf{x} + \mathbf{v}(t) \tag{5.8}$$

where $\mathbf{w}(t) \sim \mathcal{N}(0, \Sigma_w)$ is process noise, $\mathbf{x} = \begin{bmatrix} q & r & w_y & w_z \end{bmatrix}^T$, $\mathbf{y} = \begin{bmatrix} \omega_{sy} & \omega_{sz} \end{bmatrix}^T$, $\mathbf{v}(t) \sim \mathcal{N}(0, \Sigma_v)$ is sensor noise, and

$$\begin{aligned} a_{11}(t) &= \overline{q} S_{ref} D_{ref} r_{ref} c_{mqr}(t) q / I_{yy}(t) \\ a_{12}(t) &= -(I_{xx}(t) - I_{yy}(t)) pr / I_{yy}(t) \\ a_{22}(t) &= \overline{q} S_{ref} D_{ref} r_{ref} c_{mqr}(t) r / I_{yy}(t). \end{aligned}$$

5.2.1 Kalman Filter

Let Equation (5.7) be given by

$$\dot{x} = A(t)x(t) + B(t)u(t) + w(t). \tag{5.9}$$

Since the system is LTV, the states can be estimated with a time-varying Kalman filter [61]. The Kalman filter is run in discrete time so the matrices A and B must be converted to discrete time via the matrix exponential [62]. To generate time-varying model lookup tables we compute the matrix exponential off-line

$$\begin{bmatrix} F & G \\ 0 & I \end{bmatrix} = \exp\left(\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} T_s \right) \tag{5.10}$$

where T_s is sampling time. The Kalman filter then has the form

$$\begin{aligned} x[k|k] &= x[k|k-1] + L(k)(y(k) - Cx[k|k-1]) && (5.11) \\ x[k+1|k] &= F(k)x[k|k] + G(k)(u(k) + w(k)) && (5.12) \end{aligned}$$

where $F(k) \in \Re^{4\times4}$, $G(k), L(k) \in \Re^{4\times2}$ are gains calculated off-line, and $u(k)$ is the thrust provided by the control thrusters. The vector $w(k)$ represents the moment contribution due to aerodynamic forces that arise due to angle of attack and is set to zero. The off-line design procedure schedules the gains every 0.01 s for the first two seconds of flight, after which the system becomes inactive. There are a total of 32 lookup tables involved in these coefficients plus one more for estimation of pyrotechnic thruster moments.

5.3 State Estimation with a Direction Vector

In this section we expand the system model in order to permit the estimation of angular position in addition to roll, pitch, and yaw rates. The angular position is determined from a sensor that measures a vector in the body frame that is fixed in the inertial frame.

Angular position can be represented with either Euler angles or quaternions. Euler angle dynamics are given by

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s_\phi s_\theta / c_\theta & c_\phi s_\theta / c_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix} \omega$$

where s_i and c_i are $\sin(i)$ and $\cos(i)$, respectively. Quaternion dynamics are given by

$$\frac{dq}{dt} = \frac{1}{2} \begin{bmatrix} 0 & r & -q & p \\ -r & 0 & p & q \\ q & -p & 0 & r \\ -p & -q & -r & 0 \end{bmatrix} \begin{bmatrix} q_i \\ q_j \\ q_k \\ q_r \end{bmatrix}.$$

An attitude quaternion that rotates from a fixed frame to a rotating frame can be converted to corresponding Euler angles with the following equations

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{2(q_r q_i + q_j q_k)}{2(q_r^2 + q_k^2) - 1}\right) \\ \arcsin(2(q_r q_j - q_i q_k)) \\ \arctan\left(\frac{2(q_r q_k + q_i q_j)}{2(q_r^2 + q_i^2) - 1}\right) \end{bmatrix}.$$

It is no longer possible to cast the system as linear when angular position is included in the model; it is a nonlinear, time-varying system.

$$\dot{\mathbf{x}}_E = \begin{bmatrix} \frac{\overline{c}c_L(t)}{I_{xx}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\overline{c}c_{mqr}(t)}{I_{yy}(t)} & -\frac{(I_{xx}(t)-I_{yy}(t))p}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{(I_{xx}(t)-I_{yy}(t))p}{I_{yy}(t)} & \frac{\overline{c}c_{mqr}(t)}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & s_\phi s_\theta / c_\theta & c_\phi s_\theta / s_\theta & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_\phi & -s_\phi & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & c_x & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_y & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_z \end{bmatrix} \mathbf{x}_E + \begin{bmatrix} 0 & 0 \\ \frac{1}{I_{yy}(t)} & 0 \\ 0 & \frac{1}{I_{yy}(t)} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tau_{control} + \mathbf{w}(t) \quad (5.13)$$

$$\mathbf{y}_{E,Q} = \begin{bmatrix} p + w_x \\ q + w_y \\ r + w_z \\ c_\psi c_\theta H_x^I + s_\psi c_\theta H_y^I - s_\theta H_z^I \\ (c_\psi s_\theta s_\phi - s_\psi c_\phi) H_x^I + (s_\psi s_\theta s_\phi + c_\psi c_\phi) H_y^I + c_\theta s_\phi H_z^I \\ (c_\psi s_\theta c_\phi + s_\psi s_\phi) H_x^I + (s_\psi s_\theta c_\phi - c_\psi s_\phi) H_y^I + c_\theta s_\phi H_z^I \end{bmatrix} + \mathbf{v}(t) \quad (5.14)$$

or

$$
\begin{aligned}
\dot{\mathbf{x}}_E &= A(\mathbf{x},t)\mathbf{x} + B(t)\mathbf{u} + \mathbf{w}(t) \\
\mathbf{y}_{E,Q} &= h(\mathbf{x},t) + \mathbf{v}(t)
\end{aligned}
$$

where $\overline{c} = \overline{q} S_{ref} D_{ref} r_{ref}$,

$$\mathbf{x}_E = \begin{bmatrix} p & q & r & \phi & \theta & \psi & w_x & w_y & w_z \end{bmatrix}^T,$$

and

$$\mathbf{y}_{E,Q} = \begin{bmatrix} \omega_{sx} & \omega_{sy} & \omega_{sz} & H_x^B & H_y^B & H_z^B \end{bmatrix}^T + \mathbf{v}(t).$$

H_i^B is the measured direction vector in the body frame, and H_i^I is the corresponding vector in the inertial frame. Likewise, the quaternion formulation of the system is

$$\dot{\mathbf{x}}_Q = \begin{bmatrix} \frac{\overline{c}c_{Lp}(t)}{I_{xx}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\overline{c}c_{mqr}(t)}{I_{yy}(t)} & -\frac{(I_{xx}(t)-I_{yy}(t))p}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{(I_{xx}(t)-I_{yy}(t))p}{I_{yy}(t)} & \frac{\overline{c}c_{mqr}(t)}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{r}{2} & -\frac{q}{2} & \frac{p}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{r}{2} & 0 & \frac{p}{2} & \frac{q}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{q}{2} & -\frac{p}{2} & 0 & \frac{r}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{p}{2} & -\frac{q}{2} & -\frac{r}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_x & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_y & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_z \end{bmatrix} \mathbf{x}_Q + \begin{bmatrix} 0 & 0 \\ \frac{1}{I_{yy}(t)} & 0 \\ 0 & \frac{1}{I_{yy}(t)} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tau_{control} + \mathbf{w}(t) \quad (5.15)$$

$$y_{E,Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \beta & \gamma H_z^I & -\gamma H_y^I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\gamma H_z^I & \beta & \gamma H_x^I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma H_y^I & -\gamma H_x^I & \beta & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}_Q + \cos(2\cos^{-1}(q_{bi,r})) \begin{bmatrix} 0 \\ 0 \\ 0 \\ H_x^I \\ H_y^I \\ H_z^I \end{bmatrix} + \mathbf{v}(t) \quad (5.16)$$

where

$$\begin{aligned}\beta &= -\frac{1}{\sin^2(\cos^{-1}(q_{bi,r}))}(H^I_x q_{bi,i} + H^I_y q_{bi,j} + H^I_z q_{bi,k})(1 - \cos(2\cos^{-1}(q_{bi,r})))\\ \gamma &= \frac{1}{\sin(\cos^{-1}}(q_{bi,r}))\sin(2\cos^{-1}(q_{bi,r}))\end{aligned}$$

and $\mathbf{x}_Q = \begin{bmatrix} p & q & r & q_{ib,i} & q_{ib,j} & q_{ib,k} & q_{ib,r} & w_x & w_y & w_z \end{bmatrix}^T$. The quaternion subscripts bi and ib denote rotations from the body to inertial frame and inertial to body frame, respectively. These two quaternions are related by $q_{ib} = q^*_{bi}$ (see §A.2).

By using the Pythagorean theorem and some trigonometric identities, the quaternion equations can be reformulated to remove all of the trigonometric functions. Since the quaternion used in the system model is an attitude quaternion, it has a magnitude of unity or

$$q^2_{ib,i} + q^2_{ib,j} + q^2_{ib,k} + q^2_{ib,r} = q^2_{bi,i} + qbi,j^2 + q^2_{bi,k} + q^2_{bi,r} = 1. \tag{5.17}$$

This relationship is shown in Figure 5.1 where $a = \sqrt{q^2_{ib,i} + q^2_{ib,j} + q^2_{ib,k}}$ and $b = q_{ib,r}$. The results are the same for q_{bi}. By the Pythagorean theorem $\vartheta = \cos^{-1}(b) = \sin^{-1}(a)$ or $\vartheta = \cos^{-1}(q_{ib,r}) =$

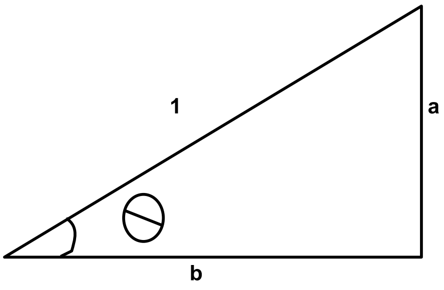

Figure 5.1: Relationship Among Rotation Quaternion Components

$\sin^{-1}\left(\sqrt{q_{ib,i}^2 + q_{ib,j}^2 + q_{ib,k}^2}\right)$. Then using the identities

$$\begin{aligned}
\sin^2(\vartheta) &= \frac{1}{2} - \frac{1}{2}\cos(2\vartheta) \\
\sin(2\vartheta) &= 2\sin(\vartheta)\cos(\vartheta) \\
\cos(2\vartheta) &= 1 - 2\sin^2(\vartheta),
\end{aligned}$$

$$\begin{aligned}
H_x^B &= \frac{1-\cos(2\vartheta}{\sin^2(\vartheta)}(q_{ib,i}H_x^I + q_{ib,j}H_y^I + q_{ib,k}H_z^I)q_{ib,i} + \frac{\sin(2\vartheta)}{\sin(\vartheta)}H_z^I q_{ib,j} - \\
&\quad \frac{\sin(2\vartheta)}{\sin(\vartheta)}H_y^I q_{ib,k} + (2\cos^2(\vartheta) - 1)H_y^I \\
H_y^B &= -\frac{\sin(2\vartheta)}{\sin(\vartheta)}H_z^I q_{ib,i} + \frac{1-\cos(2\vartheta}{\sin^2(\vartheta)}(q_{ib,i}H_x^I + q_{ib,j}H_y^I + q_{ib,k}H_z^I)q_{ib,j} + \\
&\quad \frac{\sin(2\vartheta)}{\sin(\vartheta)}H_x^I q_{ib,k} + (2\cos^2(\vartheta) - 1)H_y^I \\
H_z^B &= \frac{\sin(2\vartheta)}{\sin(\vartheta)}H_y^I q_{ib,i} - \frac{\sin(2\vartheta)}{\sin(\vartheta)}H_x^I q_{ib,j} + \\
&\quad \frac{1-\cos(2\vartheta}{\sin^2(\vartheta)}(q_{ib,i}H_x^I + q_{ib,j}H_y^I + q_{ib,k}H_z^I)q_{ib,k} + (2\cos^2(\vartheta) - 1)H_z^I
\end{aligned}$$

Then using the fact that $\sin(\theta) = \sqrt{q_{ib,i}^2 + q_{ib,j}^2 + q_{ib,k}^2}$ and $\cos(\theta) = q_{ib,r}$,

$$\begin{aligned}
H_x^B &= 2(q_{ib,i}H_x^I + q_{ib,j}H_y^I + q_{ib,k}H_z^I)q_{ib,i} + 2q_{ib,r}H_z^I q_{ib,j} + 2q_{ib,r}H_y^I q_{ib,k} - \\
&\quad (2q_{ib,r}^2 - 1)H_x^I \qquad (5.18a) \\
H_y^B &= -2q_{ib,r}H_z^I q_{ib,i} + 2(q_{ib,i}H_x^I + q_{ib,j}H_y^I + q_{ib,k}H_z^I)q_{ib,j} + 2q_{ib,r}H_x^I q_{ib,k} + \\
&\quad (2q_{ib,r}^2 - 1)H_y^I \qquad (5.18b) \\
H_z^B &= 2q_{ib,r}H_y^I q_{ib,i} - 2q_{ib,r}H_x^I q_{ib,j} + 2(q_{ib,i}H_x^I + q_{ib,j}H_y^I + q_{ib,k}H_z^I)q_{ib,k} + \\
&\quad (2q_{ib,r}^2 - 1)H_z^I \qquad (5.18c)
\end{aligned}$$

Then

$$\gamma = 2q_{ib,r} \tag{5.19a}$$

$$\beta = 2(q_{ib,i}H_x^I + q_{ib,j}H_y^I + q_{ib,k}H_z^I). \tag{5.19b}$$

The output equation is thus

$$\mathbf{y}_{E,Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \beta & \gamma H_z^I & -\gamma H_y^I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\gamma H_z^I & \beta & \gamma H_x^I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma H_y^I & -\gamma H_x^I & \beta & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}_Q + (2q_{ib,r}^2 - 1) \begin{bmatrix} 0 \\ 0 \\ 0 \\ H_x^I \\ H_y^I \\ H_z^I \end{bmatrix} + \mathbf{v}(t). \tag{5.20}$$

Extended Kalman Filter (EKF)

Implementation of an EKF requires the availability of a linearized model of the nonlinear systems given by Equations (5.13)-(5.14) and Equations (5.15)-(5.16). To linearize the output equations, partial derivatives of Equations (5.18a)-(5.18c) are necessary

$$\frac{\partial H_x^B}{\partial q_{ib,i}} = 2q_{ib,i}H_x^I + 2(q_{ib,i}H_x^I + q_{ib,j}H_y^I + q_{ib,k}H_z^I) \tag{5.21}$$

$$\frac{\partial H_x^B}{\partial q_{ib,j}} = 2q_{ib,i}H_y^I + 2q_{ib,r}H_z^I \tag{5.22}$$

$$\frac{\partial H_x^B}{\partial q_{ib,k}} = 2q_{ib,i}H_z^I - 2q_{ib,r}H_y^I \tag{5.23}$$

$$\frac{\partial H_x^B}{\partial q_{ib,r}} = 2q_{ib,j}H_z^I - 2q_{ib,k}H_y^I + 4q_{ib,r}H_x^I \tag{5.24}$$

$$\begin{aligned}
\frac{\partial H_y^B}{\partial q_{ib,i}} &= -2q_{ib,r}H_z^I + 2q_{ib,j}H_x^I && (5.25)\\
\frac{\partial H_y^B}{\partial q_{ib,j}} &= 2q_{ib,j}H_y^I + 2(q_{ib,i}H_x^I + q_{ib,j}H_y^I + q_{ib,k}H_z^I) && (5.26)\\
\frac{\partial H_y^B}{\partial q_{ib,k}} &= 2q_{ib,j}H_z^I + 2q_{ib,r}H_x^I && (5.27)\\
\frac{\partial H_y^B}{\partial q_{ib,r}} &= -2q_{ib,i}H_z^I + 2q_{ib,k}H_x^I + 4q_{ib,r}H_y^I && (5.28)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial H_z^B}{\partial q_{ib,i}} &= 2q_{ib,r}H_y^I + 2q_{ib,k}H_x^I && (5.29)\\
\frac{\partial H_z^B}{\partial q_{ib,j}} &= -2q_{ib,r}H_x^I + 2q_{ib,k}H_y^I && (5.30)\\
\frac{\partial H_z^B}{\partial q_{ib,k}} &= 2q_{ib,k}H_z^I + 2(q_{ib,i}H_x^I + q_{ib,j}H_y^I + q_{ib,k}H_z^I) && (5.31)\\
\frac{\partial H_z^B}{\partial q_{ib,r}} &= 2q_{ib,i}H_y^I - 2q_{ib,j}H_x^I + 4q_{ib,r}H_z^I. && (5.32)
\end{aligned}$$

Equation (5.5) can be written as

$$\dot{\omega} = \begin{bmatrix} f_1(x,t) \\ f_2(x,t) \\ f_3(x,t) \end{bmatrix} + B(t)\tau_{control}. \tag{5.33}$$

$B(t)$ is already linear so no action is necessary. The remainder of the equation is linearized as follows

$$\begin{array}{lll}
\frac{\partial f_1}{\partial p} = \overline{c}c_{Lp}(t)/I_{xx}(t) & \frac{\partial f_1}{\partial q} = 0 & \frac{\partial f_1}{\partial r} = 0 \\
\frac{\partial f_2}{\partial p} = -\frac{(I_{xx}(t)-I_{yy}(t))r}{I_{yy}(t)} & \frac{\partial f_2}{\partial q} = \overline{c}c_{mqr}(t)/I_{yy}(t) & \frac{\partial f_2}{\partial r} = -\frac{(I_{xx}(t)-I_{yy}(t))p}{I_{yy}(t)} \\
\frac{\partial f_3}{\partial p} = -\frac{(I_{xx}(t)-I_{yy}(t))q}{I_{yy}(t)} & \frac{\partial f_3}{\partial q} = -\frac{(I_{xx}(t)-I_{yy}(t))p}{I_{yy}(t)} & \frac{\partial f_3}{\partial r} = \overline{c}c_{mqr}(t)/I_{yy}(t).
\end{array}$$

Then the system model becomes

$$
\begin{aligned}
\dot{\mathbf{x}}_Q &= \begin{bmatrix}
\frac{\overline{c}c_{Lp}(t)}{I_{xx}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\frac{(I_{xx}(t)-I_{yy}(t))}{I_{yy}(t)}r & \frac{\overline{c}c_{mqr}(t)}{I_{yy}(t)} & -\frac{(I_{xx}(t)-I_{yy}(t))}{I_{yy}(t)}p & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\frac{(I_{xx}(t)-I_{yy}(t))}{I_{yy}(t)}q & -\frac{(I_{xx}(t)-I_{yy}(t))}{I_{yy}(t)}r & \frac{\overline{c}c_{mqr}(t)}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{r}{2} & -\frac{q}{2} & \frac{p}{2} & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{r}{2} & 0 & \frac{p}{2} & \frac{q}{2} & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{q}{2} & -\frac{p}{2} & 0 & \frac{r}{2} & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{p}{2} & -\frac{q}{2} & -\frac{r}{2} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & c_x & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_y & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_z
\end{bmatrix} \mathbf{x}_Q + \begin{bmatrix}
0 & 0 \\
\frac{1}{I_{yy}(t)} & 0 \\
0 & \frac{1}{I_{yy}(t)} \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0
\end{bmatrix} \tau_{control} \\
& \quad +\mathbf{w}(t) \\
\mathbf{y}_{E,Q} &= \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & \frac{\partial H_x^B}{\partial q_{ib,i}} & \frac{\partial H_x^B}{\partial q_{ib,j}} & \frac{\partial H_x^B}{\partial q_{ib,k}} & \frac{\partial H_x^B}{\partial q_{ib,r}} & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{\partial H_y^B}{\partial q_{ib,i}} & \frac{\partial H_y^B}{\partial q_{ib,j}} & \frac{\partial H_y^B}{\partial q_{ib,k}} & \frac{\partial H_y^B}{\partial q_{ib,r}} & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{\partial H_z^B}{\partial q_{ib,i}} & \frac{\partial H_z^B}{\partial q_{ib,j}} & \frac{\partial H_z^B}{\partial q_{ib,k}} & \frac{\partial H_z^B}{\partial q_{ib,r}} & 0 & 0 & 0
\end{bmatrix} \mathbf{x}_Q + \mathbf{v}(t)
\end{aligned}
$$

or

$$\begin{aligned}\dot{\mathbf{x}}_Q &= A_E\mathbf{x}_Q + B\mathbf{u} + \mathbf{w} \\ \mathbf{y}_{E,Q} &= C_E\mathbf{x}_Q + \mathbf{v}\end{aligned}$$

where $\mathbf{v}$ is zeros for the elements corresponding to the ideal vector output. The matrices A_E, B, and C_E are formed by evaluating each entry at a specific time value and the state estimate at the previous time step.

5.3.1 Ideal Vector

One method to estimate the rocket's attitude is to have a sensor that measures a vector in the body frame that is known in the inertial frame. The measurement taken in the body frame can be rotated to match the the known vector in the inertial frame to determine the rocket's attitude. In this section the ideal case where a known unit vector $H^I = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$ pointing in the x-direction in the inertial frame is considered. Although any known vector in the inertial frame is valid, the x-direction unit vector is used for simplicity.

With $H^I_x = 1$ and $H^I_y = H^I_z = 0$, Equations (5.18a)-(5.18c) become

$$\begin{aligned}H^B_x &= 2q^2_{ib,i} + 2q^2_{ib,r} - 1 \\ H^B_y &= 2q_{ib,i}q_{ib,j} + 2q_{ib,k}q_{ib,r} \\ H^B_z &= 2q_{ib,i}q_{ib,k} - 2q_{ib,r}q_{ib,j}.\end{aligned}$$

Then the coefficients γ and β of the system output $y_{E,Q}$ (Equation (5.20)) are

$$\begin{aligned}\gamma &= 2q_{ib,r} \\ \beta &= 2q_{ib,i}.\end{aligned}$$

The output equation $y_{E,Q}$ is then

$$y_{E,Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \beta & 0 & 0 & \gamma & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \beta & \gamma & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\gamma & \beta & 0 & 0 & 0 & 0 \end{bmatrix} x_Q - \begin{bmatrix} 0 \\ 0 \\ 0 \\ H_x^I \\ H_y^I \\ H_z^I \end{bmatrix}.$$

The linearized equations used for an EKF are then

$$\begin{array}{llll} \frac{\partial H_x^B}{\partial q_{ib,i}} = 4q_{ib,i} & \frac{\partial H_x^B}{\partial q_{ib,j}} = 0 & \frac{\partial H_x^B}{\partial q_{ib,k}} = 0 & \frac{\partial H_x^B}{\partial q_{ib,r}} = 4q_{ib,r} \\ \frac{\partial H_y^B}{\partial q_{ib,i}} = 2q_{ib,j} & \frac{\partial H_y^B}{\partial q_{ib,j}} = 2q_{ib,i} & \frac{\partial H_y^B}{\partial q_{ib,k}} = 2q_{ib,r} & \frac{\partial H_y^B}{\partial q_{ib,r}} = 2q_{ib,k} \\ \frac{\partial H_z^B}{\partial q_{ib,i}} = -2q_{ib,k} & \frac{\partial H_z^B}{\partial q_{ib,j}} = 2q_{ib,r} & \frac{\partial H_z^B}{\partial q_{ib,k}} = -2q_{ib,i} & \frac{\partial H_z^B}{\partial q_{ib,r}} = 2q_{ib,j}. \end{array}$$

5.3.2 Magnetometer Measurements

The ideal vector situation represents the best possible scenario, but a magnetometer gives realistic performance. The Earth's magnetic field is fixed in the inertial frame, so a magnetometer measuring the field in the body frame can be used to determine attitude. However, the Earth's magnetic field is not constant, as discussed in Chapter 3. In this section, for the short time intervals of the estimation process, the magnetic field can be approximated as being constant. The magnetic field is determined by taking measurements of the magnetic field with a magnetometer attached to the rocket prior to the launch.

The magnetometer sensor readings are modeled as

$$H_{si}^B = H_i^B + w_{mi} + b_{mi} + \eta_{mi} \tag{5.34}$$

where $i = \{x, y, z\}$, η_i is zero-mean white noise, and b_i is a constant bias. w_{mi} is a walking bias modeled as a first order Markov process with dynamics

$$\dot{w}_{mi} = c_{mi} w_{mi} + \eta_i. \tag{5.35}$$

Then with a magnetometer and rate gyros, the system model is

$$
\begin{aligned}
\dot{\mathbf{x}}_E &= \begin{bmatrix}
\frac{\overline{c}c_L(t)}{I_{xx}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{\overline{c}c_{mqr}(t)}{I_{yy}(t)} & -\frac{(I_{xx}(t)-I_{yy}(t))p}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{(I_{xx}(t)-I_{yy}(t))p}{I_{yy}(t)} & \frac{\overline{c}c_{mqr}(t)}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & s_\phi s_\theta / c_\theta & c_\phi s_\theta / s_\theta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & c_\phi & -s_\phi & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & s_\phi / c_\theta & c_\phi / c_\theta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & c_x & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & c_y & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_z & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{mx} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{my} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{mz}
\end{bmatrix} \mathbf{x}_E + \begin{bmatrix}
0 & 0 \\
\frac{1}{I_{yy}(t)} & 0 \\
0 & \frac{1}{I_{yy}(t)} \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0
\end{bmatrix} \tau_{control} \\
&+ \mathbf{w}(t)
\end{aligned}
$$

$$\mathbf{y}_{E,Q} = \begin{bmatrix} p + w_x \\ q + w_y \\ r + w_z \\ c_\psi c_\theta H_x^I + s_\psi c_\theta H_y^I - s_\theta H_z^I + w_{mx} \\ (c_\psi s_\theta s_\phi - s_\psi c_\phi)H_x^I + (s_\psi s_\theta s_\phi + c_\psi c_\phi)H_y^I + c_\theta s_\phi H_z^I + w_{my} \\ (c_\psi s_\theta c_\phi + s_\psi s_\phi)H_x^I + (s_\psi s_\theta c_\phi - c_\psi s_\phi)H_y^I + c_\theta s_\phi H_z^I + w_{mz} \end{bmatrix}^T + \mathbf{v}(t)$$

where

$$\mathbf{x}_E = \begin{bmatrix} p & q & r & \phi & \theta & \psi & w_x & w_y & w_z & w_{mx} & w_{my} & w_{mz} \end{bmatrix}^T$$

and

$$\mathbf{y}_{E,Q} = \begin{bmatrix} \omega_{sx} & \omega_{sy} & \omega_{sz} & H_{sx}^B & H_{sy}^B & H_{sz}^B \end{bmatrix}^T + \mathbf{v}(t)$$

for the Euler angle formulation. For the quaternion formulation the system is

$$
\begin{aligned}
\dot{\mathbf{x}}_Q &= \begin{bmatrix}
\frac{\bar{c}c_{Lp}(t)}{I_{xx}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{\bar{c}c_{mqr}(t)}{I_{yy}(t)} & -\frac{(I_{xx}(t)-I_{yy}(t))p}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{(I_{xx}(t)-I_{yy}(t))p}{I_{yy}(t)} & \frac{\bar{c}c_{mqr}(t)}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{r}{2} & -\frac{q}{2} & \frac{p}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{r}{2} & 0 & \frac{p}{2} & \frac{q}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{q}{2} & -\frac{p}{2} & 0 & \frac{r}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{p}{2} & -\frac{q}{2} & -\frac{r}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & c_x & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_y & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_z & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{mx} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{my} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{mz}
\end{bmatrix} \mathbf{x}_Q + \qquad (5.36) \\
& \begin{bmatrix}
0 & \frac{1}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}^T \tau_{control} + \mathbf{w}(t) \qquad (5.37)
\end{aligned}
$$

$$\mathbf{y}_{E,Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta & \gamma H_z^I & -\gamma H_y^I & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\gamma H_z^I & \beta & \gamma H_x^I & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \gamma H_y^I & -\gamma H_x^I & \beta & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_Q + (2q_{ib,r}^2 - 1) \begin{bmatrix} 0 & 0 & 0 & H_x^I & H_y^I & H_z^I \end{bmatrix}^T + \mathbf{v}(t) \tag{5.38}$$

where $\mathbf{x}_Q = \begin{bmatrix} p & q & r & q_{ib,i} & q_{ib,j} & q_{ib,k} & q_{ib,r} & w_x & w_y & w_z & w_{mx} & w_{my} & w_{mz} \end{bmatrix}^T$ and γ and β are given by Equations (5.19a) and (5.19b).

5.4 Estimation with Rate and Angle Gyros

This section develops estimators for the rocket when the onboard sensors are rate gyros and gyros that sense angular position directly. A rate gyro can be used to estimate angular position by integrating the sensor output. But due to noise on the sensor output, integration causes the angular position estimates to become increasingly inaccurate in a matter of milliseconds. Shkel and Painter [63], [64], [65] have proposed a type of MEMS gyroscope (referred to in this paper as an angle gyro) that directly measures rotational angles, thus eliminating the problem of integrating noise associated with a typical MEMS gyro (referred to hence forward as a rate gyro). The angle gyro is still in the developmental stages; however, it is still a MEMS device, and its output can be modeled in the same manner as a rate gyro. A gyro that senses angular position directly avoids the problem of integrating noise, thereby producing more accurate readings. Such gyro readings are modeled as

$$
\begin{aligned}
\phi_s &= \phi + w_\phi + b_\phi + \eta_\phi && (5.39)\\
\theta_s &= \theta + w_\theta + b_\theta + \eta_\theta && (5.40)\\
\psi_s &= \psi + w_\psi + b_\psi + \eta_\psi && (5.41)
\end{aligned}
$$

where ϕ, θ, and ψ are Euler angles, b_i are constant biases, and η_i are zero-mean Gaussian noise. w_i is a walking bias modeled as a first-order Markov process with dynamics

$$\dot{w}_i = c_i w_i + \eta_i. \tag{5.42}$$

The system can be modeled in terms of Euler angles as follows. The state vector is given by

$$\mathbf{x}_E = \begin{bmatrix} p & q & r & \phi & \theta & \psi & w_x & w_y & w_z & w_\phi & w_\theta & w_\psi \end{bmatrix}^T,$$

and the output vector is given by

$$\mathbf{y}_{E,Q} = \begin{bmatrix} \omega_{sx} & \omega_{sy} & \omega_{sz} & \phi_s & \theta_s & \psi_s \end{bmatrix}^T + \mathbf{v}(t).$$

Then the system model is

$$\dot{\mathbf{x}}_E = \begin{bmatrix} \frac{\overline{c}c_{Lp}(t)}{I_x x(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\overline{c}c_{mqr}(t)}{I_{yy}(t)} & -\frac{(I_{xx}(t)-I_{yy}(t))p}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\overline{c}c_{mqr}(t)}{I_{yy}(t)} & \frac{(I_{xx}(t)-I_{yy}(t))p}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & \frac{s_\phi s_\theta}{c_\theta} & \frac{c_\phi s_\theta}{c_\theta} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_\phi & -s_\phi & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{s_\phi}{c_\theta} & \frac{c_\phi}{c_\theta} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & c_x & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_y & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_\phi & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_\theta & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_\psi \end{bmatrix} \mathbf{x}_E + \begin{bmatrix} 0 & 0 \\ \frac{1}{I_{yy}(t)} & 0 \\ 0 & \frac{1}{I_{yy}(t)} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tau_{control} + \mathbf{w}(t) \qquad (5.43)$$

$$\mathbf{y}_{E,Q} = \begin{bmatrix} I_{6\times 6} & I_{6\times 6} \end{bmatrix} \mathbf{x}_E + \mathbf{v}(t). \tag{5.44}$$

The system dynamics can be formulated in terms of quaternions as follows. The state vector is given by

$$x_Q = \begin{bmatrix} p & q & r & q_{ib,i} & q_{ib,j} & q_{ib,k} & q_{ib,r} & w_x & w_y & w_z & w_\phi & w_\theta & w_\psi \end{bmatrix}^T,$$

and the output vector is the same as for the Euler formulation. The system model is then

$$\dot{\mathbf{x}}_Q = \begin{bmatrix} \frac{\overline{c}c_{Lp}(t)}{I_x x(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\overline{c}c_{mqr}(t)}{I_{yy}(t)} & -\frac{(I_{xx}(t)-I_{yy}(t))p}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\overline{c}c_{mqr}(t)}{I_{yy}(t)} & \frac{(I_{xx}(t)-I_{yy}(t))p}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{r}{2} & -\frac{q}{2} & \frac{p}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{r}{2} & 0 & \frac{p}{2} & \frac{q}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{q}{2} & -\frac{p}{2} & 0 & \frac{r}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{p}{2} & -\frac{q}{2} & -\frac{r}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_x & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_y & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_\phi & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_\theta & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_\psi \end{bmatrix} \mathbf{x}_Q + \begin{bmatrix} 0 & 0 \\ \frac{1}{I_{yy}(t)} & 0 \\ 0 & \frac{1}{I_{yy}(t)} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tau_{control}$$

$$+\mathbf{w}(t) \tag{5.45}$$

$$\mathbf{y}_{E,Q} = \begin{bmatrix} p + w_x \\ q + w_y \\ r + w_z \\ \tan^{-1}\left(\frac{2(q_{ib,j}q_{ib,k}+q_{ib,i}q_{ib,r})}{2(q_{ib,r}^2+q_{ib,k}^2)-1}\right) \\ \sin^{-1}(2(q_{ib,r}q_{ib,j} - q_{ib,i}q_{ib,k})) \\ \tan^{-1}\left(\frac{2(q_{ib,r}q_{ib,k}+q_{ib,i}q_{ib,j})}{2(q_{ib,r}^2+q_{ib,i}^2)-1}\right) \end{bmatrix} + \mathbf{v}(t). \tag{5.46}$$

5.4.1 EKF

Implementation of an EKF requires the linearization of the system model given Equations (5.43) and (5.45). The Euler angle dynamic equations are

$$\begin{aligned} \dot{\phi} &= p + \frac{\sin(\phi)\sin(\theta)}{\cos(\theta)}q + \frac{\cos(\phi)\sin(\theta)}{\cos(\theta)}r \\ \dot{\theta} &= \cos(\theta)q - \sin(\phi)r \\ \dot{\psi} &= \frac{\sin(\phi)}{\cos(\theta)}q + \frac{\cos(\phi)}{\cos(\theta)}r. \end{aligned}$$

The associated partial derivatives are

$$\begin{array}{lll} \frac{\partial\dot{\phi}}{\partial p} = 1 & \frac{\partial\dot{\phi}}{\partial q} = \frac{\sin(\phi)\sin(\theta)}{\cos(\theta)} & \frac{\partial\dot{\phi}}{\partial r} = \frac{\cos(\phi)\sin(\theta)}{\cos(\theta)} \\ \frac{\partial\dot{\phi}}{\partial\phi} = \frac{\cos(\phi)\sin(\theta)}{\cos(\theta)}q - \frac{\sin(\phi)\sin(\theta)}{\cos(\theta)}r & \frac{\partial\dot{\phi}}{\partial\theta} = \frac{\sin(\phi)}{\cos^2(\theta)}q + \frac{\cos(\phi)}{\cos^2(\theta)}r & \frac{\partial\dot{\phi}}{\partial\psi} = 0 \\ \frac{\partial\dot{\theta}}{\partial p} = 0 & \frac{\partial\dot{\theta}}{\partial q} = \cos(\theta) & \frac{\partial\dot{\theta}}{\partial r} = -\sin(\phi) \\ \frac{\partial\dot{\theta}}{\partial\phi} = -\cos(\phi)r & \frac{\partial\dot{\theta}}{\partial\theta} = -\sin(\theta)q & \frac{\partial\dot{\theta}}{\partial\psi} = 0 \\ \frac{\partial\dot{\psi}}{\partial p} = 0 & \frac{\partial\dot{\psi}}{\partial q} = \frac{\sin(\phi)}{\cos(\theta)} & \frac{\partial\dot{\psi}}{\partial r} = \frac{\cos(\phi)}{\cos(\theta)} \\ \frac{\partial\dot{\psi}}{\partial\phi} = \frac{\cos(\phi)}{\cos(\theta)}q - \frac{\sin(\phi)}{\cos(\theta)}r & \frac{\partial\dot{\psi}}{\partial\theta} = \frac{\sin(\phi)\sin(\theta)}{\cos^2(\theta)}q + \frac{\cos(\phi)\sin(\theta)}{\cos^2(\theta)}r & \frac{\partial\dot{\psi}}{\partial\psi} = 0 \end{array}$$

For the quaternion formulation, f_1, f_2, and f_3 of Equation (5.33) are

$$\begin{aligned} f_1 &= \tan^{-1}\left(\frac{2(q_{ib,j}q_{ib,k} + q_{ib,i}q_{ib,r})}{2(q_{ib,r}^2 + q_{ib,k}^2) - 1}\right) && (5.47)\\ f_2 &= \sin^{-1}(2(q_{ib,r}q_{ib,j} - q_{ib,i}q_{ib,k})) && (5.48)\\ f_3 &= \tan^{-1}\left(\frac{2(q_{ib,r}q_{ib,k} + q_{ib,i}q_{ib,j})}{2(q_{ib,r}^2 + q_{ib,i}^2) - 1}\right). && (5.49) \end{aligned}$$

Since

$$\frac{d\tan^{-1}(x)}{dx} = \frac{1}{1+x^2}$$

and

$$\frac{d\sin^{-1}(x)}{dx} = \frac{1}{\sqrt{1-x^2}},$$

the partial derivatives of Equations (5.47)-(5.49)are developed as follows. The partial derivative of f_1 with respect to $q_{ib,r}$ is

$$\begin{aligned} \frac{\partial f_1}{\partial q_{ib,r}} &= \frac{\left(2q_{ib,i}(2(q_{ib,r}^2 + q_{ib,k}^2 - 1)) - 4q_{ib,r}(-q_{ib,r}q_{ib,i} + q_{ib,j}q_{ib,k})\right)}{\left(1 + \frac{4(-q_{ib,r}q_{ib,i}+q_{ib,j}q_{ib,k})^2}{(2(q_{ib,r}^2+q_{ib,k}^2-1)^2}\right)(2(q_{ib,r}^2 + q_{ib,k}^2) - 1)^2}\\ &= \frac{2q_{ib,i}(2(q_{ib,r}^2 + q_{ib,k}^2) - 1) - 8q_{ib,r}(q_{ib,j}q_{ib,k} + q_{ib,r}q_{ib,i})}{(2(q_{ib,r}^2 + q_{ib,k}^2) - 1)^2 + 4(q_{ib,j}q_{ib,k} + q_{ib,r}q_{ib,i})^2}. \end{aligned}$$

The denominator of $\frac{\partial f_1}{\partial q_{ib,r}}$ can be rewritten as

$$\begin{aligned} &(2(q_{ib,r}^2 + q_{ib,k}^2) - 1)^2 + 4(q_{ib,j}q_{ib,k} - q_{ib,r}q_{ib,i})^2\\ &= 4(q_{ib,r}^2 + q_{ib,k}^2)(q_{ib,r}^2 + q_{ib,k}^2) - 4(q_{ib,r}^2 + q_{ib,k}^2) + 1 + 4q_{ib,j}^2q_{ib,k}^2 +\\ &\quad 8q_{ib,i}q_{ib,j}q_{ib,k}q_{ib,r} + 4q_{bi,r}^2q_{bi,i}^2\\ &= 4q_{ib,r}^4 + 4q_{ib,k}^4 + 8q_{ib,r}^2q_{ib,k}^2 - 4q_{ib,r}^2 - 4q_{ib,k}^2 + 1 + 4q_{ib,j}^2q_{ib,k}^2 +\\ &\quad 8q_{ib,i}q_{ib,j}q_{ib,k}q_{ib,r} + 4q_{ib,r}^2q_{ib,i}^2\\ &= 1 - 4(q_{ib,r}q_{ib,j} - q_{ib,i}q_{ib,k})^2. \end{aligned}$$

Then

$$\frac{\partial f_1}{\partial q_{ib,r}} = \frac{4q_{ib,i}(q_{ib,k}^2 - q_{ib,r}^2) - 8q_{ib,j}q_{ib,k}q_{ib,r} - 2q_{ib,i}}{1 - 4(q_{ib,r}q_{ib,j} - q_{ib,i}q_{ib,k})^2}. \tag{5.50}$$

The remaining derivatives are

$$\begin{aligned}
\frac{\partial f_1}{\partial q_{ib,i}} &= \frac{4q_{ib,r}(q_{ib,r}^2 + q_{ib,k}^2) - 2q_{ib,r}}{1 - 4(q_{ib,r}q_{ib,j} - q_{ib,i}q_{ib,k})^2} && (5.51)\\
\frac{\partial f_1}{\partial q_{ib,j}} &= \frac{-4q_{ib,k}(q_{ib,r}^2 + q_{ib,k}^2) - 2q_{ib,k}}{1 - 4(q_{ib,r}q_{ib,j} - q_{ib,i}q_{ib,k})^2} && (5.52)\\
\frac{\partial f_1}{\partial q_{ib,k}} &= \frac{4q_{ib,j}(q_{ib,r}^2 - q_{ib,k}^2) - 8q_{ib,i}q_{ib,k}q_{ib,r} - 2q_{ib,i}}{1 - 4(q_{ib,r}q_{ib,j} - q_{ib,i}q_{ib,k})^2} && (5.53)\\
\frac{\partial f_2}{\partial q_{ib,r}} &= \frac{2q_{ib,j}}{\sqrt{1 - 4(q_{ib,r}q_{ib,j} - q_{ib,i}q_{ib,k})^2}} && (5.54)\\
\frac{\partial f_2}{\partial q_{ib,i}} &= \frac{-2q_{ib,k}}{\sqrt{1 - 4(q_{ib,r}q_{ib,j} - q_{ib,i}q_{ib,k})^2}} && (5.55)\\
\frac{\partial f_2}{\partial q_{ib,j}} &= \frac{2q_{ib,r}}{\sqrt{1 - 4(q_{ib,r}q_{ib,j} - q_{ib,i}q_{ib,k})^2}} && (5.56)\\
\frac{\partial f_2}{\partial q_{ib,k}} &= \frac{-2q_{ib,i}}{\sqrt{1 - 4(q_{ib,r}q_{ib,j} - q_{ib,i}q_{ib,k})^2}}. && (5.57)
\end{aligned}$$

With respect to $q_{ib,r}$ the partial derivative of f_3 is

$$\frac{\partial f_3}{\partial q_{ib,r}} = \frac{4q_{ib,k}(q_{ib,r}^2 + q_{ib,i}^2) - 2q_{ib,k} - 8q_{ib,r}(q_{ib,i}q_{ib,j} - q_{ib,r}q_{ib,k})}{(2(q_{ib,r}^2 + q_{ib,i}^2) - 1)^2 + 4(q_{ib,i}q_{ib,j} - q_{ib,r}q_{ib,k})^2}.$$

By the same method used to calculate the denominator of $\frac{\partial f_1}{\partial q_{ib,r}}$, it can be shown that

$$(2(q_{ib,r}^2 + q_{ib,i}^2) - 1)^2 + 4(q_{ib,i}q_{ib,j} - q_{ib,r}q_{ib,k})^2 = 1 - 4(q_{ib,r}q_{ib,j} - q_{ib,i}q_{ib,k})^2.$$

Then

$$\frac{\partial f_3}{\partial q_{ib,r}} = \frac{4q_{ib,k}(q_{ib,r}^2 - q_{ib,i}^2) - 8q_{ib,r}q_{ib,i}q_{ib,j} - 2q_{ib,k}}{1 - 4(q_{ib,r}q_{ib,j} - q_{ib,i}q_{ib,k})^2} \tag{5.58}$$

$$\frac{\partial f_3}{\partial q_{ib,i}} = \frac{4q_{ib,j}(q_{ib,r}^2 - q_{ib,i}^2) - 8q_{ib,r}q_{ib,i}q_{ib,k} - 2q_{ib,j}}{1 - 4(q_{ib,r}q_{ib,j} - q_{ib,i}q_{ib,k})^2} \tag{5.59}$$

$$\frac{\partial f_3}{\partial q_{ib,j}} = \frac{4q_{ib,i}(q_{ib,r}^2 + q_{ib,i}^2) - 2q_{ib,i}}{1 - 4(q_{ib,r}q_{ib,j} - q_{ib,i}q_{ib,k})^2} \tag{5.60}$$

$$\frac{\partial f_3}{\partial q_{ib,k}} = \frac{4q_{ib,r}(q_{ib,r}^2 + q_{ib,i}^2) - 2q_{ib,r}}{1 - 4(q_{ib,r}q_{ib,j} - q_{ib,i}q_{ib,k})^2} \tag{5.61}$$

5.5 Estimation with Magnetometer, Rate Gyros, and Angle Gyros

In this section estimators are developed for the rocket when the sensors used are a magnetometer, rate gyros, and angle gyros. The sensor models are the same as used in the previous sections. Linearizations derived in previous sections are also used in developing the EKF in this section. By using both angle sensing gyros and a magnetometer, two sensors are being used to determine angular position.

For the Euler formulation the state vector is

$$\mathbf{x}_E = \begin{bmatrix} p & q & r & \phi & \theta & \psi & w_x & w_y & w_z & w_\phi & w_\theta & w_\psi & w_{mx} & w_{my} & w_{mz} \end{bmatrix}^T.$$

The output vector is

$$\mathbf{y}_{E,Q} = \begin{bmatrix} \omega_{sx} & \omega_{sy} & \omega_{sz} & \phi_s & \theta_s & \psi_s & H_x^B & H_y^B & H_z^B \end{bmatrix}^T + \mathbf{v}(t).$$

The system is then given by

$$
\dot{\mathbf{x}}_E = \begin{bmatrix}
\frac{\overline{c}c_{Lp}(t)}{I_x x(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{\overline{c}c_{mqr}(t)}{I_{yy}(t)} & -\frac{(I_{xx}(t)-I_{yy}(t))p}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{\overline{c}c_{mqr}(t)}{I_{yy}(t)} & \frac{(I_{xx}(t)-I_{yy}(t))p}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & \frac{s_\phi s_\theta}{c_\theta} & \frac{c_\phi s_\theta}{c_\theta} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & c_\phi & -s_\phi & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{s_\phi}{c_\theta} & \frac{c_\phi}{c_\theta} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & c_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & c_y & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_z & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_\phi & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_\theta & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_\psi & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{mx} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{my} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{mz}
\end{bmatrix} \mathbf{x}_E +
\begin{bmatrix}
0 & \frac{1}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}^T \tau_{control} + \mathbf{w}(t)
$$

$$\mathbf{y}_{E,Q} = \begin{bmatrix} p + w_x & q + w_y & r + w_z & \phi + w_\phi & \theta + w_\theta & \psi + w_\psi & y_7 & y_8 & y_9 \end{bmatrix}^T + \mathbf{v}(t)$$

where

$$\begin{aligned} y_7 &= \cos(\psi)\cos(\theta)H_x^I + \sin(\psi)\cos(\theta)H_y^I - \sin(\theta)H_x^I \\ y_8 &= (\cos(\psi)\sin(\theta)\sin(\psi) - \sin(\psi)\cos(\phi))H_x^I + (\sin(\psi)\sin(\theta)\sin(\phi) + \\ &\quad \cos(\psi)\cos(\phi))H_y^I + \cos(\theta)\sin(\phi)H_z^I \\ y_9 &= (\cos(\psi)\sin(\theta)\cos(\phi) + \sin(\psi)\sin(\phi))H_x^I + (\sin(\psi)\sin(\theta)\cos(\phi) - \\ &\quad \cos(\psi)\sin(\phi))H_y^I + \cos(\theta)\cos(\phi)H_z^I \end{aligned}$$

The quaternion formulation state vector is

$$\mathbf{x}_Q = \begin{bmatrix} p & q & r & q_{ib,i} & q_{ib,j} & q_{ib,k} & q_{ib,r} & w_x & w_y & w_z & w_\phi & w_\theta & w_\psi & w_{mx} & w_{my} & w_{mz} \end{bmatrix}.$$

The output vector is

$$\mathbf{y}_{E,Q} = \begin{bmatrix} \omega_{sx} & \omega_{xy} & \omega_{sz} & \phi_s & \theta_s & \psi_s & H_x^I & H_y^I & H_z^I \end{bmatrix} + \mathbf{v}(t).$$

The system dynamics are then

$$
\begin{aligned}
\dot{\mathbf{x}}_Q \quad = & \left[\begin{array}{cccccccccccccccc}
\frac{\overline{c}c_{Lp}(t)}{I_{xx}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{\overline{c}c_{mqr}(t)}{I_{yy}(t)} & -\frac{(I_{xx}(t)-I_{yy}(t))p}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{(I_{xx}(t)-I_{yy}(t))p}{I_{yy}(t)} & \frac{\overline{c}c_{mqr}(t)}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{r}{2} & -\frac{q}{2} & \frac{p}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{r}{2} & 0 & \frac{p}{2} & \frac{q}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{q}{2} & -\frac{p}{2} & 0 & \frac{r}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{p}{2} & -\frac{q}{2} & -\frac{r}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & c_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_y & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_z & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_\phi & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_\theta & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_\psi & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{mx} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{my} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{mz}
\end{array}\right] \mathbf{x}_Q + \\
& \left[\begin{array}{cccccccccccccccc}
0 & \frac{1}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]^T \tau_{control} + \mathbf{w}(t) \qquad (5.62)
\end{aligned}
$$

$$\begin{aligned}\mathbf{y} &= \Big[\ p+w_x \quad q+w_y \quad r+w_z \quad f_1+w_\phi \quad f_2+w_\theta \quad f_3+w_\psi \quad H_x^B+w_{mx} \\ &\quad H_y^B+w_{my} \quad H_z^B+w_{mz}\ \Big]+\mathbf{v}(t)\end{aligned} \tag{5.63}$$

where H_x^B, H_y^B, and H_z^B are given by Equations (5.18a)-(5.18c) and f_1, f_2, and f_3 are given by Equations (5.47)-(5.49).

5.5.1 EKF

For the Euler formation these additional linearizations are necessary for an EKF:

$$\begin{aligned}
\frac{\partial y_7}{\partial \phi} &= 0 \\
\frac{\partial y_7}{\partial \theta} &= -\sin(\theta)(\cos(\psi)H_x^I+\sin(\psi)H_y^I+H_z^I) \\
\frac{\partial y_7}{\partial \psi} &= \cos(\theta)(-\sin(\psi)H_x^I+\cos(\psi)H_y^I) \\
\frac{\partial y_8}{\partial \phi} &= (\cos(\psi)\sin(\theta)\cos(\phi)+\sin(\psi)\sin(\phi))H_x^I+(\sin(\psi)\sin(\theta)\cos(\psi)- \\
&\quad \cos(\psi)\sin(\phi))H_y^I+\cos(\theta)\cos(\phi)H_z^I \\
\frac{\partial y_8}{\partial \theta} &= \cos(\psi)\cos(\theta)\sin(\phi)H_x^I+\sin(\psi)\cos(\theta)\sin(\phi)H_y^I-\sin(\theta)\sin(\phi)H_z^I \\
\frac{\partial y_8}{\partial \psi} &= (-\sin(\psi)\sin(\theta)\sin(\phi)-\cos(\psi)\cos(\phi))H_x^I+(\cos(\psi)\sin(\theta)\sin(\phi)- \\
&\quad \sin(\psi)\cos(\phi))H_y^I \\
\frac{\partial y_9}{\partial \phi} &= (-\cos(\psi)\sin(\theta)\sin(\phi)+\sin(\psi)\cos(\phi))H_x^I+(-\sin(\psi)\sin(\theta)\sin(\phi)- \\
&\quad \cos(\psi)\cos(\phi))H_y^I-\cos(\theta)\sin(\phi)H_z^I \\
\frac{\partial y_9}{\partial \theta} &= \cos(\psi)\cos(\theta)\cos(\phi)H_x^I-\sin(\psi)\cos(\theta)\cos(\phi)H_y^I-\sin(\theta)\cos(\phi)H_z^I \\
\frac{\partial y_9}{\partial \psi} &= (-\sin(\psi)\sin(\theta)\cos(\phi)+\cos(\psi)\sin(\phi))H_x^I+(\cos(\psi)\sin(\theta)\cos(\phi)+ \\
&\quad \sin(\psi)\sin(\phi))H_y^I.
\end{aligned}$$

For the quaternion formulation, the necessary linearizations for the fourth through sixth terms of y are given by Equations (5.50)-(5.61) and for H_x^B, H_y^B, and H_z^B by Equations (5.21)-(5.32).

5.6 Wahba's Problem Applied to a Rocket

In this section we develop the equations necessary to use a Wahba's problem solution for state estimation of a rotating rocket. As presented in Chapter 2, it is possible to get an estimate of attitude if two noncolinear vectors that are measured in the rocket's body frame are known in the inertial frame. For satellites these two measured vectors are usually a magnetic field vector (measured with a magnetometer) and a gravitational field vector (measured with an accelerometer). However, an accelerometer cannot be used to measure the gravitational field vector of a rocket in flight.

To get the two needed vector sets to solve Wahba's problem, we use a magnetometer and angle gyros. The angle gyros measure the rocket's rotational angles, which can then be used to rotate a vector known in the inertial frame to the rocket's body frame. We rotate the inertial vector $\frac{1}{\sqrt{3}}\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$ into the body frame. With two sets of vectors from the magnetometer measurements and angle gyro readings, we apply the ESOQ algorithm (see §2.13) to estimate the rocket's attitude quaternion. The ESOQ algorithm is chosen because it provides a closed-form solution in the form of a quaternion to the attitude estimation problem at a low computational cost. Estimates of the rotational rates can be found via Euler differentiation by first converting the quaternion to a set of Euler angles. Then subtract the Euler angles of the current time step's Euler angles from the Euler angles of the previous time step, and divide by the time step.

The remainder of this section is organized as follows. In §5.6.1 we develop a model for using the ESOQ algorithm in conjunction with an EKF and a UKF. The ESOQ algorithm provides an estimate of the attitude quaternion from magnetometer and angle gyro measurements. The quaternion is then provided as a sensor reading to the system model. Since the ESOQ algorithm provides a quaternion solution, Euler angle formations for the systems are omitted in §5.6.1. Finally, in §5.6.2, we present a model using the ESOQ algorithm to estimate angular position and Kalman filtering to estimate angular rates.

5.6.1 Wahba's Problem Solution as Input to an EKF and UKF

In this section we develop the system model when the solution to Wahba's problem (found via the ESOQ algorithm) is modeled as a sensor measurement for an EKF and a UKF. The rocket is assumed to be equipped with a magnetometer, rate gyros, and angle gyros as in §5.5. The rate gyro and magnetometer measurements are inputs to the ESOQ algorithm, which estimates the rocket's attitude quaternion. The estimated quaternion along with measurements from the rate gyros are used as sensor readings for the EKF and UKF.

The state vector x_Q for the system is

$$x_Q = \begin{bmatrix} p & q & r & q_{ib,i} & q_{ib,j} & q_{ib,k} & q_{ib,r} & w_x & w_y & w_z & w_i & w_j & w_k & w_r \end{bmatrix}^T$$

where w_i, w_j, w_k, and w_r are walking biases associated with the quaternion estimate from the ESOQ solution with dynamics

$$\dot{w}_n = c_n w_n + \eta_n \qquad n = \{i, j, k, r\}.$$

The system dynamics are then

$$
\begin{aligned}
\dot{\mathbf{x}}_Q \quad = & \begin{bmatrix}
\frac{\overline{c}c_{Lp}(t)}{I_x x(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{\overline{c}c_{mqr}(t)}{I_{yy}(t)} & -\frac{(I_{xx}(t)-I_{yy}(t))p}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{(I_{xx}(t)-I_{yy}(t))p}{I_{yy}(t)} & \frac{\overline{c}c_{mqr}(t)}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{r}{2} & -\frac{q}{2} & \frac{p}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{r}{2} & 0 & \frac{p}{2} & \frac{q}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{q}{2} & -\frac{p}{2} & 0 & \frac{r}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{p}{2} & -\frac{q}{2} & -\frac{r}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & c_x & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_y & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_z & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_i & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_j & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_k & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_r
\end{bmatrix} \mathbf{x}_Q \\
& + \begin{bmatrix}
0 & \frac{1}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{I_{yy}(t)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}^T \tau_{control} + \mathbf{w}(t)
\end{aligned} \tag{5.64}
$$

.

The output equation y is

$$\begin{aligned} y &= \begin{bmatrix} p+w_x & q+w_y & r+w_z & q_{ib,i}+w_i & q_{ib,j}+w_j & q_{ib,k}+w_k & q_{ib,r}+w_r \end{bmatrix}^T \\ &+ \mathbf{v}(t) \qquad (5.65) \\ &= \begin{bmatrix} I_{7\times 7} & I_{7\times 7} \end{bmatrix} x_Q + \mathbf{v}(t). \end{aligned}$$

5.6.2 Wahba's Problem Solution Combined with an EKF and UKF

In this section we develop the system model when the ESOQ algorithm and Kalman filtering are used independently. The rocket's attitude is estimated using the ESOQ algorithm, and an EKF or UKF is used to estimate angular rates. The state vector x_Q is

$$x_Q = \begin{bmatrix} p & q & r & w_x & w_y & w_z \end{bmatrix}^T.$$

The system dynamics are then

$$\begin{aligned} \dot{\mathbf{x}}_Q &= \begin{bmatrix} \frac{\bar{c}c_{Lp}(t)}{I_xx(t)} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\bar{c}c_{mqr}(t)}{I_{yy}(t)} & -\frac{(I_{xx}(t)-I_{yy}(t))p}{I_{yy}(t)} & 0 & 0 & 0 \\ 0 & \frac{(I_{xx}(t)-I_{yy}(t))p}{I_{yy}(t)} & \frac{\bar{c}c_{mqr}(t)}{I_{yy}(t)} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_x & 0 & 0 \\ 0 & 0 & 0 & 0 & c_y & 0 \\ 0 & 0 & 0 & 0 & 0 & c_z \end{bmatrix} \mathbf{x}_Q + \\ & \begin{bmatrix} 0 & \frac{1}{I_{yy}(t)} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_{yy}(t)} & 0 & 0 & 0 \end{bmatrix}^T \tau_{control} + \mathbf{w}(t). \end{aligned} \qquad (5.66)$$

The output equation y is given by

$$y = \begin{bmatrix} p + w_x & q + w_y & r + w_z \end{bmatrix}^T + \mathbf{v}(t) \tag{5.67}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} x_Q + \mathbf{v}(t). \tag{5.68}$$

Chapter 6
Results

In this chapter we present the results of simulating the flight of a rotating rocket as modeled in Chapter 5 when the missile's control law is provided state estimates from a KF, an EKF, a UKF [66] (see Appendix C.3 for a discussion of the UKF), and the ESOQ algorithm. We simulate the rocket with various sensor suites and compare the performances of the various estimators. In §6.1 we examine estimator performance when the rocket is equipped only with rate gyros. Next, in §6.2 we simulate both rate gyros and an ideal sensor that perfectly measures a known inertial frame vector in the body frame, and we compare the EKF and UKF performances. In §6.3 we replace the ideal sensor of §6.2 with a magnetometer and analyze estimator performance. In §6.4 we compare the performance of the EKF and the UKF when the available sensors are rate gyros and angle gyros, which directly measure the rocket's attitude. Next in §6.5 we compare the performance of the ESOQ algorithm to estimators based on Kalman filtering. Finally in §6.6, we summarize our findings and draw some conclusions.

Monte-Carlo simulations of a spinning rocket are used to compare the performance of the EKF, UKF, and ESOQ for estimation of angular rates and position. 250 simulations are run with varied parameters, which include rocket mass properties, main motor misalignment, wind disturbances, the time at which the tail fins deploy, and tip-off error (angular rates experienced immediately after exit from the launcher). The various estimators are run in tandem for each simulation. The method we use to evaluate system performance is circular error probability (CEP), the radius of the circle in which the rocket impacts 50% of the time. To calculate CEP exactly is difficult, so several approximations have been developed. The third method proposed by Taub and Thomas

[67] is used in this work:

$$
\begin{aligned}
CEP &= \left(2\chi^2_{\nu,0.5}/\nu\right)^{1/2}\left(\frac{\sigma_x^2+\sigma_y^2}{2}\right)^{1/2} && (6.1a)\\
\nu &= \frac{\left(\sigma_x^2+\sigma_y^2\right)^2}{\sigma_x^4+\sigma_y^4} && (6.1b)
\end{aligned}
$$

where σ_x^2 is cross range variance, σ_y^2 is down range variance, and $\chi^2_{\nu,0.5}$ is the chi-squared distribution with ν degrees of freedom. We also use the average of the mean squared error (MSE) over all Monte Carlo runs of rocket angular rates and position to evaluate performance. For the CEP all values are normalized, and for MSE all values are scaled by dividing by a maximum value.

6.1 Rate Gyros

In this section we compare the KF (implemented for the system model given by Equations (5.7)-(5.8) with the method of §5.2.1) and UKF (implemented with Algorithm C.3.1 for the system model given by Equation (5.4)) performances when rate gyros are the only sensors used for state estimation. In Figures 6.1-6.3 and 6.4-6.6 we show the average MSE of the KF and UKF estimates of rotational rates and angular position. The KF average MSEs are plotted in blue, and the UKF average MSEs are plotted in green. As we show in Figure 6.1, the KF estimates of roll rate p have an MSE around 0 while the UKF estimates have an MSE of almost 1. However, we see from Figures 6.2 and 6.3 that the KF and UKF have comparable average MSEs for estimates of pitch rate q and yaw rate r. The UKF average MSE spikes at approximately 1 s while the KF estimates remain fairly smooth.

In Figures 6.4-6.6 we see the MSEs of the KF and UKF estimates of roll ϕ, pitch θ, and yaw ψ angles. The MSE of the roll angle oscillates for both the KF and the UKF with the UKF settling to a smaller MSE than the KF after 1.5 s. The KF and UKF have almost identical MSEs for pitch as shown in Figure 6.5. The KF MSE of yaw angle is almost 0 while the UKF MSE oscillates between 0 and 1.

We show normalized CEP plotted versus time in Figures 6.7 and 6.8. The baseline (uncontrolled) CEP is shown with a blue line, the controlled CEP is shown with a green line, and the difference between the uncontrolled and controlled CEP is shown with a red line. In Figure 6.7 we show the CEP when the KF provides estimates to the rocket's control law, and in Figure 6.8 we show the CEP when the UKF provides estimates to the rocket's control law. Controlling the rocket with the KF as its estimator lowers the CEP by more than 9 units while the UKF only lowers CEP by a little less than 4 units.

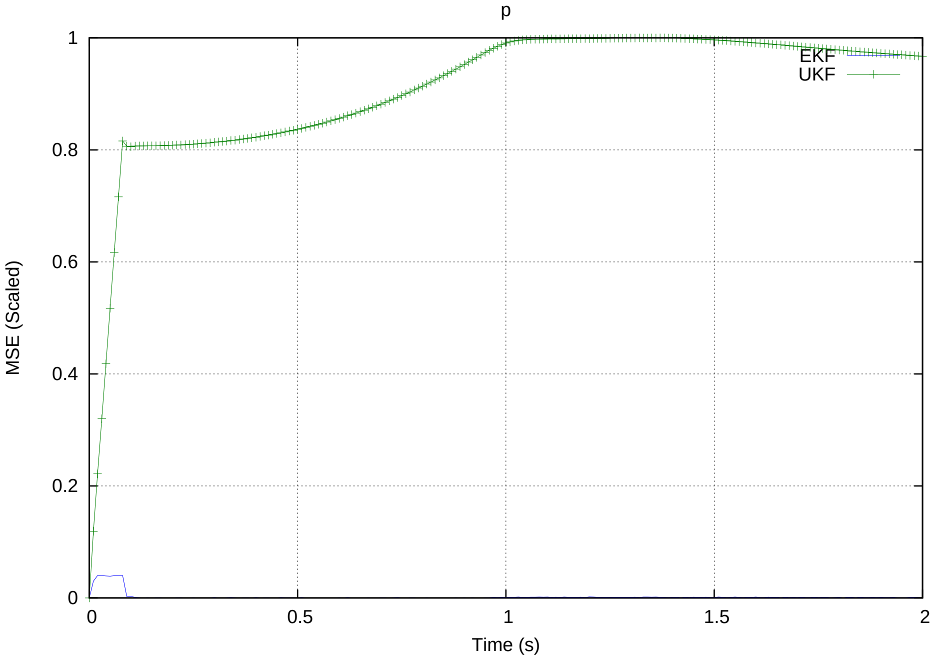

Figure 6.1: Scaled MSE of Roll Rates of KF and UKF Estimates from Rate Gyros

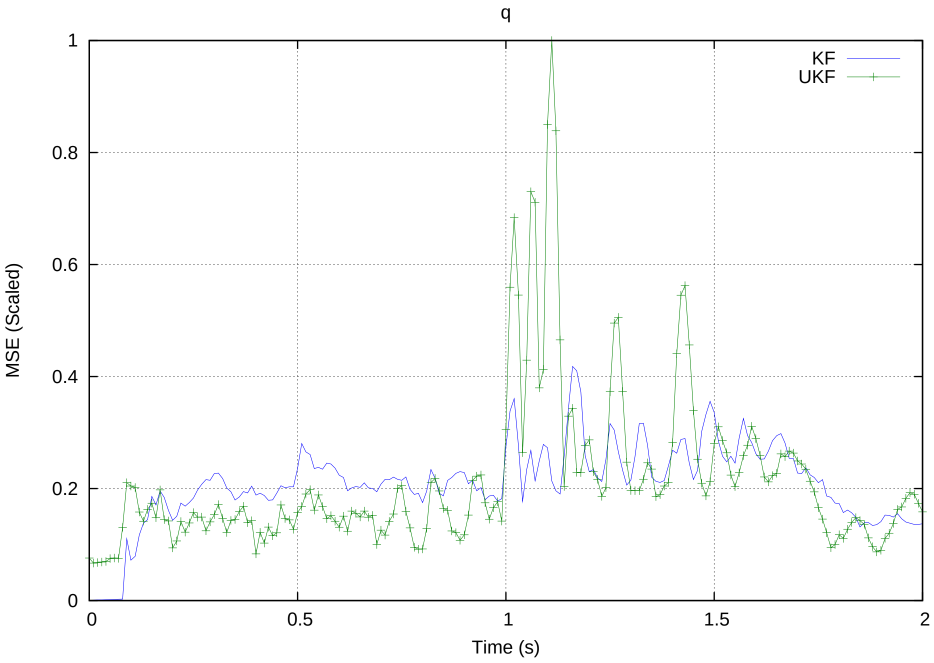

Figure 6.2: Scaled MSE of Pitch Rates of KF and UKF Estimates from Rate Gyros

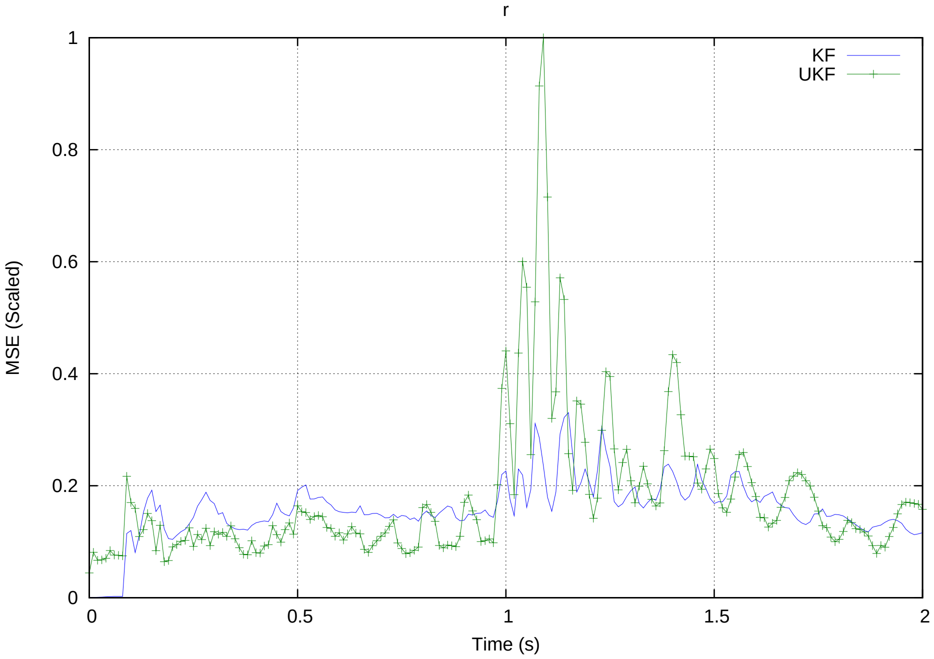

Figure 6.3: Scaled MSE of Yaw Rates of KF and UKF Estimates from Rate Gyros

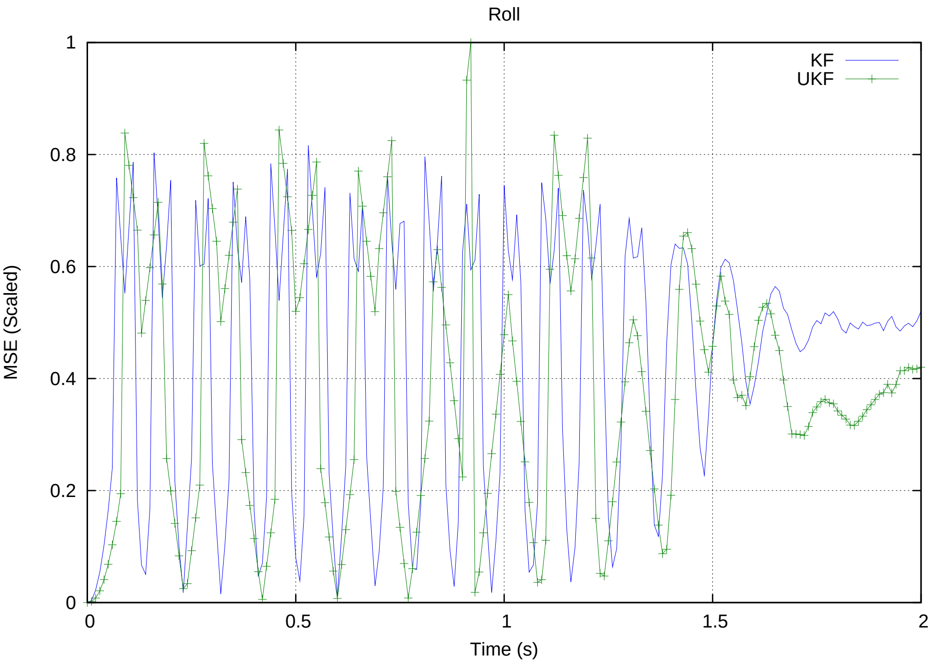

Figure 6.4: Scaled MSE of Roll Angles of KF and UKF Estimates from Rate Gyros

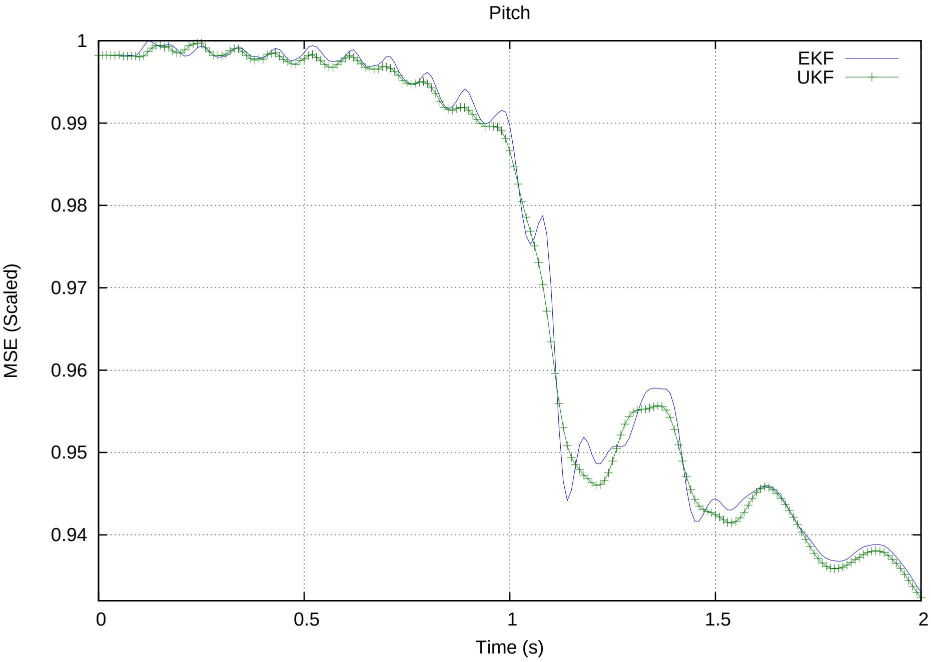

Figure 6.5: Scaled MSE of Pitch Angles of KF and UKF Estimates from Rate Gyros

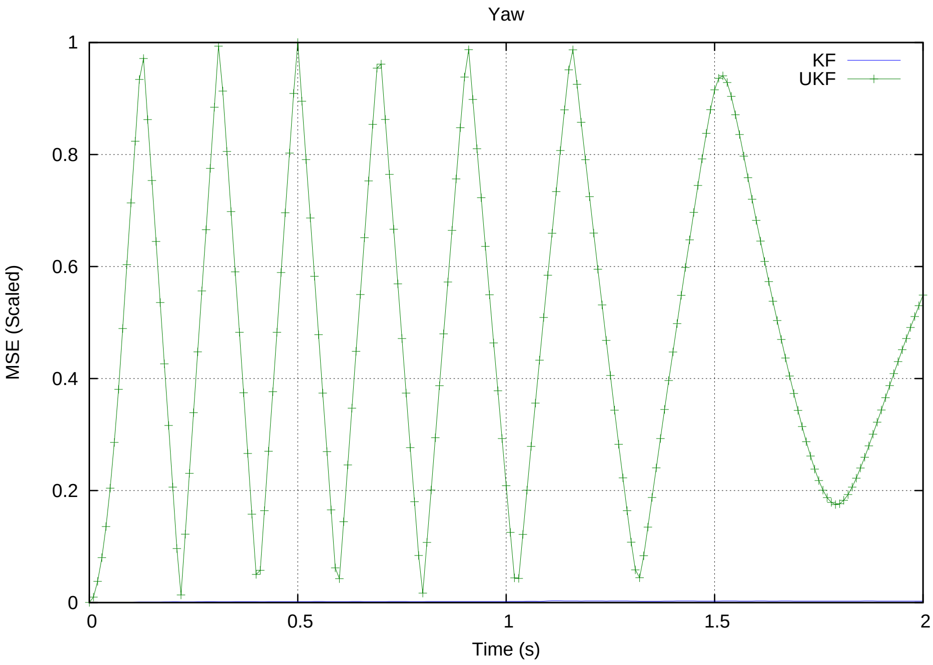

Figure 6.6: Scaled MSE of Yaw Angles of KF and UKF Estimates from Rate Gyros

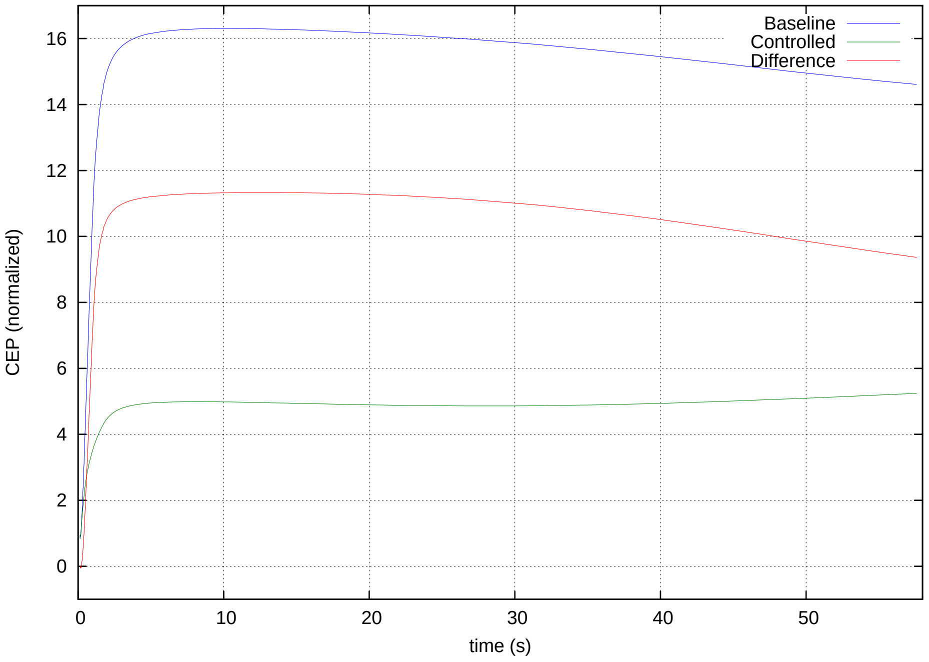

Figure 6.7: Normalized CEP for KF Estimates from Rate Gyros

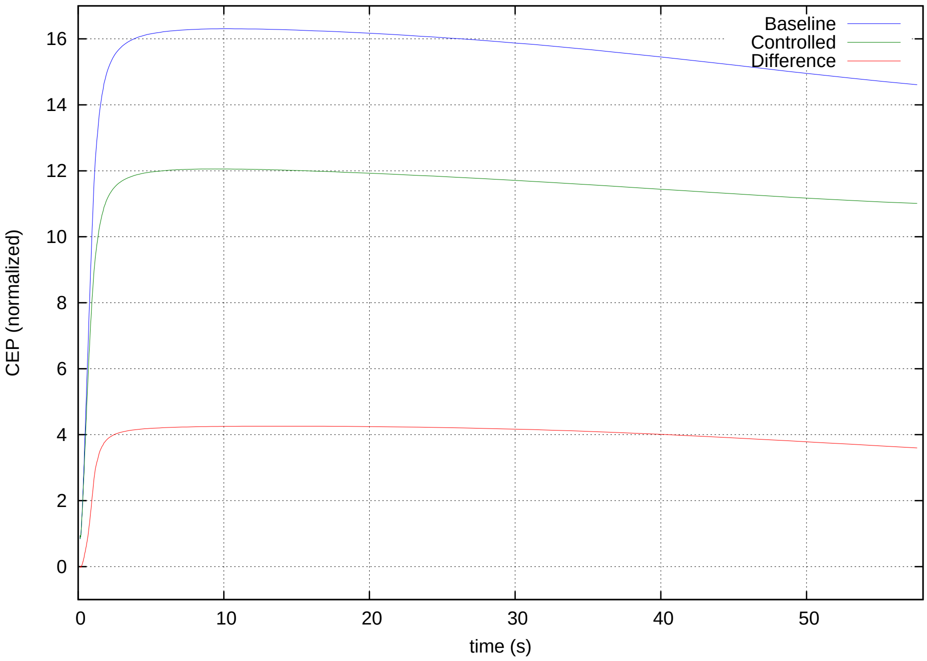

Figure 6.8: Normalized CEP for UKF Estimates from Rate Gyros

6.2 Rate Gyros and an Ideal Vector

In this section we compare the EKF (implemented with Algorithm C.2.1 for the system model given by Equations (5.15) and (5.20) using the equations developed in §5.3) and UKF (implemented with Algorithm C.3.1 for the system model given by Equations (5.15) and (5.20)) performances when rate gyros and an ideal vector sensor are used for state estimation (for details of model development see Chapter 5 and details of algorithm implementation see C). In Figures 6.9-6.11 and 6.12-6.14 we show the average MSEs of the EKF and UKF estimates of rotational rates and angular position. The EKF average MSEs are plotted in blue, and the UKF average MSEs are plotted in green. As we see from Figure 6.9, the UKF roll rate MSE starts at 0, increases to almost 0.6, and approaches 0 within the first half second. The MSE jumps to 1 and then steadily decreases. The EKF roll rate MSE increases quickly from 0 to more than 0.2 and then more slowly increases to 0.3 until 1 s at which point it remains constant. The EKF MSE is consistently smaller than the UKF MSE after 0.5 s. In Figures 6.10 and 6.11 we see that the EKF provides better initial estimates of the pitch and yaw rate, and then the UKF average MSE approaches that of the EKF.

As shown in Figure 6.12, the EKF MSE for roll angle oscillates between 0 and 1 with an average value of around 0.5. The UKF oscillates about 0.5 between 0.3 and 1 and settles to 0.6. The UKF MSE for pitch angle as shown in 6.13 oscillates until reaching steady state at around 1.75 s. The EKF MSE exhibits the same behavior, but while the EKF MSE oscillations have the same phase as the UKF MSE, the magnitude is larger. In Figure 6.14 it is shown that both the EKF and UKF MSE oscillate, but the EKF MSE oscillates between 0.5 and 1 for the first second while the UKF MSE oscillates between 0 and 1. The UKF MSE settles to just under 0.6 at about 1.75 s while the EKF MSE does not settle within 2 s.

In Figure 6.15 we show the CEP resulting from the EKF, and in Figure 6.16 we show the CEP resulting from the UKF. The baseline case is shown in blue, the controlled case is shown in green, and the difference between the baseline and controlled cases is shown in red. The EKF reduces the base line CEP from about 15 units to almost 6 units for a difference of about 9 units. The UKF only reduces the baseline CEP by a little over 1 unit.

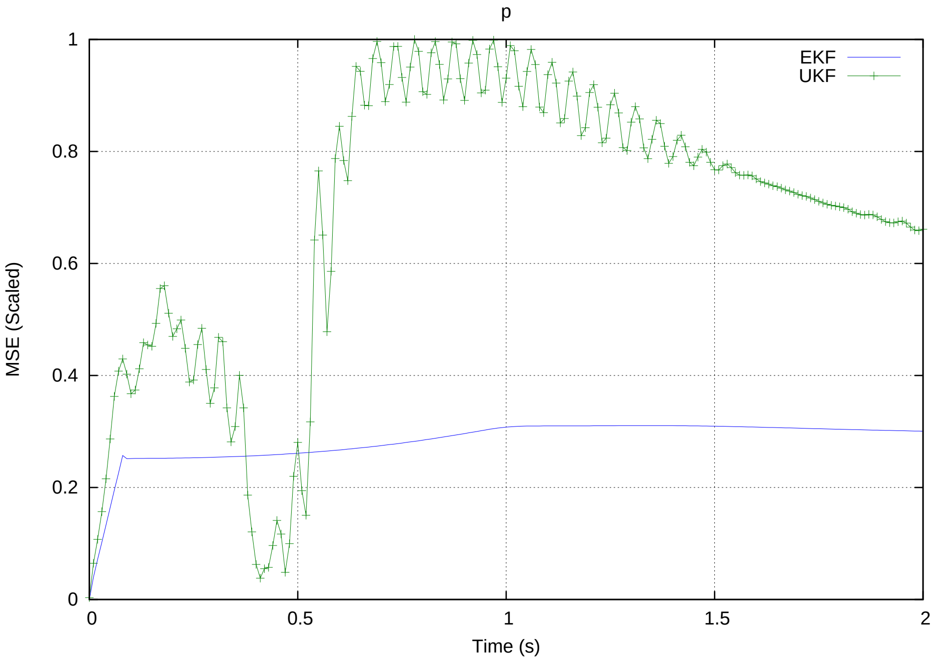

Figure 6.9: Scaled MSE of Roll Rates of EKF and UKF Estimates from Rate Gyros and an Ideal Vector

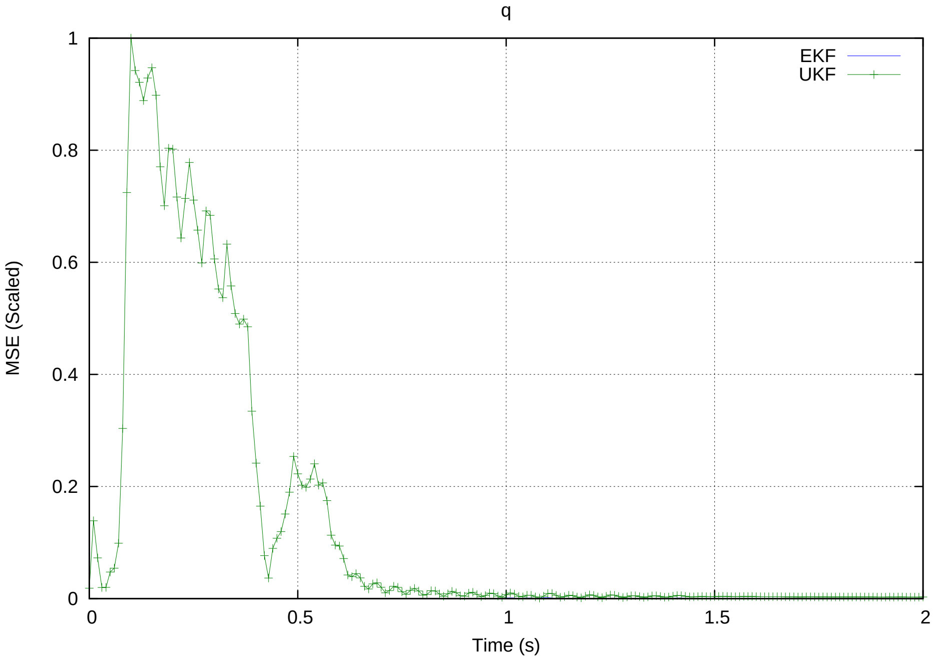

Figure 6.10: Scaled MSE of Pitch Rates of EKF and UKF Estimates from Rate Gyros and an Ideal Vector

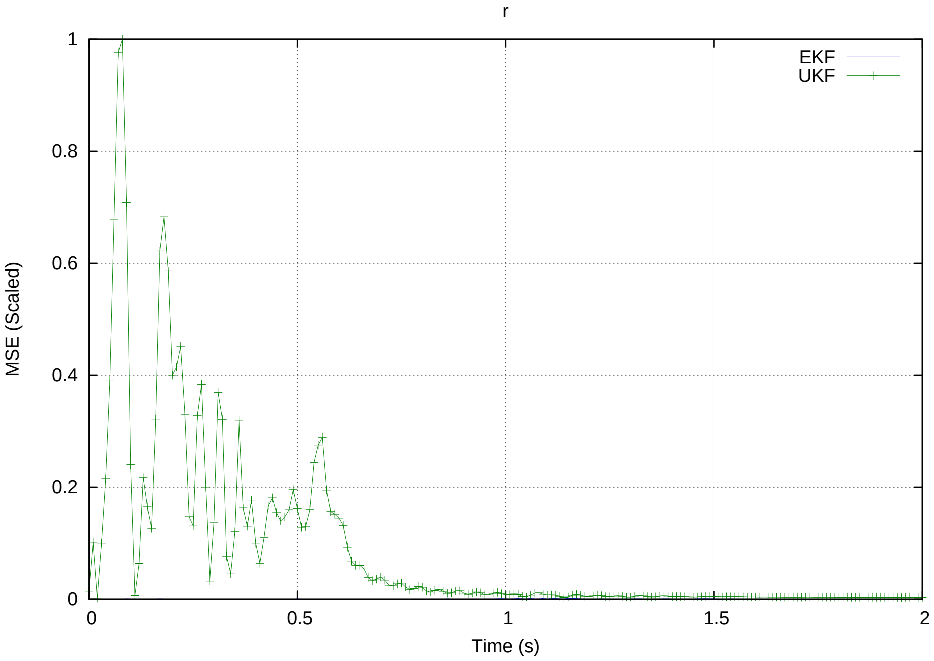

Figure 6.11: Scaled MSE of Yaw Rates of EKF and UKF Estimates from Rate Gyros and an Ideal Vector

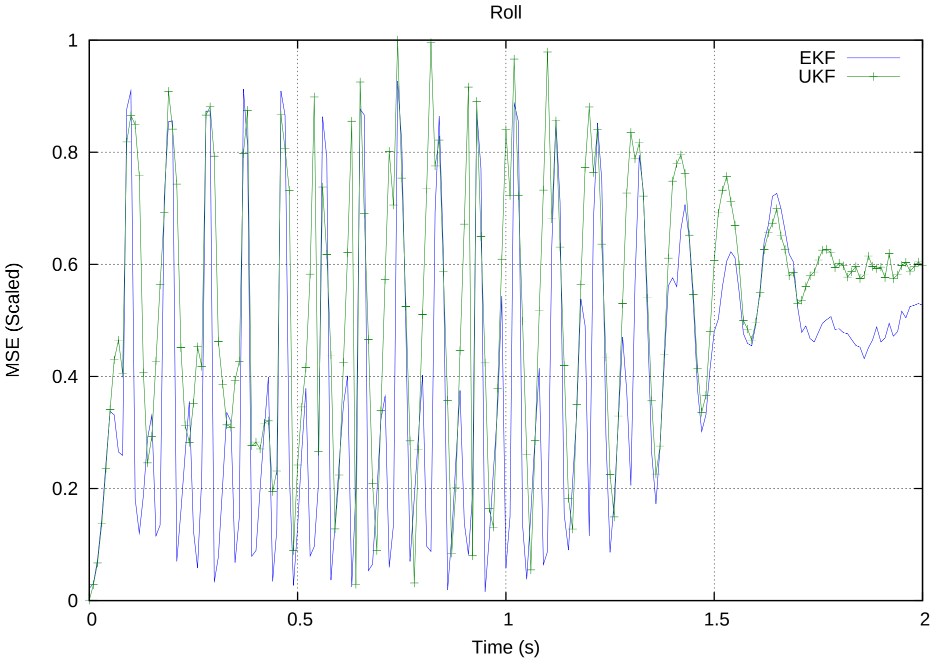

Figure 6.12: Scaled MSE of Roll Angles of EKF and UKF Estimates from Rate Gyros and an Ideal Vector

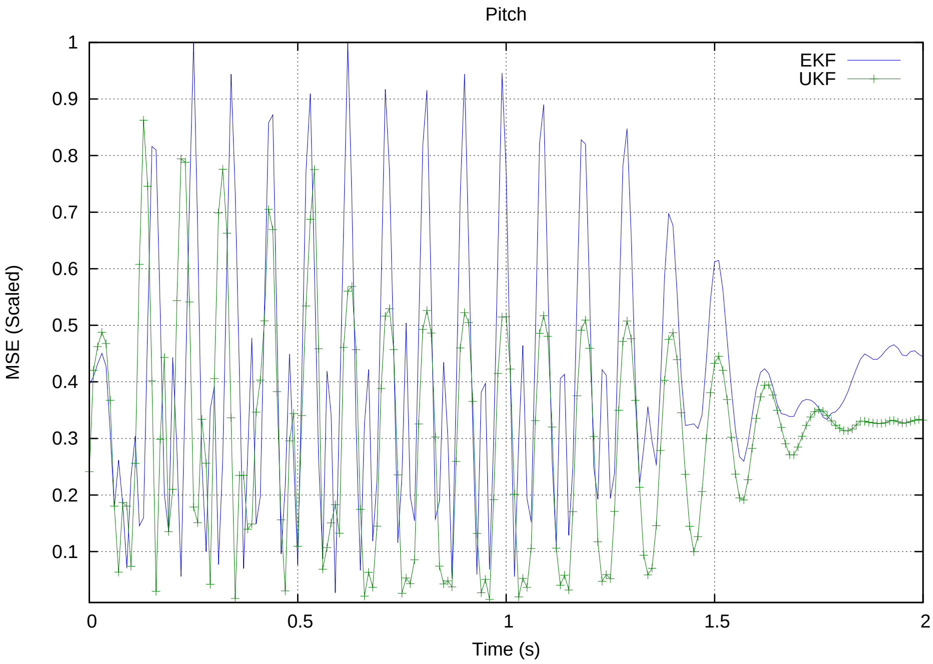

Figure 6.13: Scaled MSE of Pitch Angles of EKF and UKF Estimates from Rate Gyros and an Ideal Vector

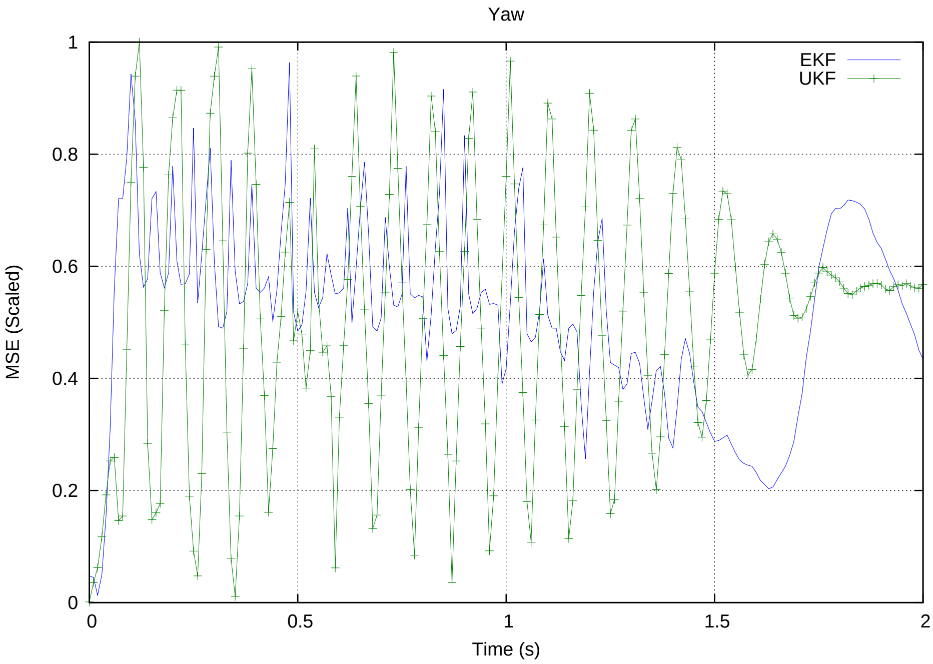

Figure 6.14: Scaled MES of Yaw Angles of EKF and UKF Estimates from Rate Gyros and an Ideal Vector

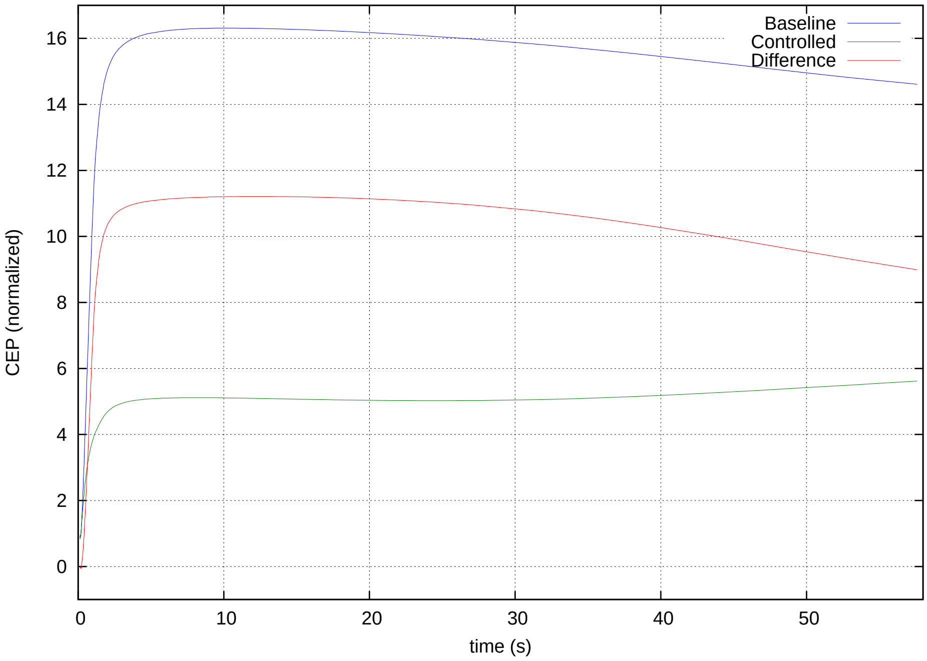

Figure 6.15: Normalized CEP for EKF Estimates from Rate Gyros and an Ideal Vector

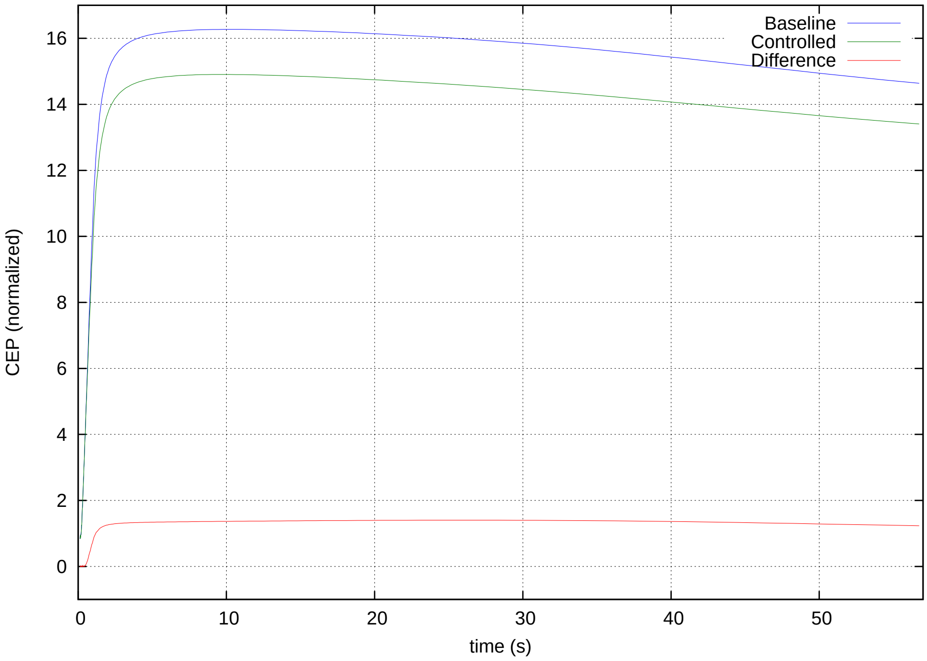

Figure 6.16: Normalized CEP for UKF Estimates from Rate Gyros and an Ideal Vector

6.3 Rate Gyros and a Magnetometer

In this section we compare the EKF (implemented with Algorithm C.2.1 for the system model given by Equations (5.37) and (5.38) using the equations developed in §5.3) and UKF (implemented with Algorithm C.3.1 for the system model given by Equations (5.37) and (5.38)) performances when rate gyros and a magnetometer are used for state estimation. In Figures 6.17-6.19 and 6.20-6.22, we show the average MSE of EKF and UKF estimates of rotational rates and angular position, respectively. EKF average MSEs are plotted in blue, and UKF average MSEs are plotted in green. As we see in Figure 6.17, the EKF provides estimates of roll rate with the same average MSE as the UKF. For pitch and yaw rate, the EKF and UKF MSEs are almost the same with no significant differences, as shown in Figures 6.18 and 6.19.

In Figures 6.20-6.22 we show the average MSE of the roll ϕ, pitch θ, and yaw ψ angles, respectively. The roll angle MSE for the EKF and UKF both oscillate between 0 and 0.8 but have opposing phases. The UKF pitch angle MSE holds almost constant at a little over 0.3 units while the EKF MSE oscillates approximately about 0.4 before it settles to about that value. The UKF MSE for yaw angle oscillates between the 0 and 1. The EKF MSE oscillates about 0.6 faster than the UKF MSE before reaching steady state.

The MSE plots of pitch rate and yaw rate estimates in Figures 6.10, 6.18, 6.11, and 6.19 explain why the ideal vector case improves the CEP less than the magnetometer case for UKF estimates. The MSE of the UKF estimates spikes almost immediately for the ideal vector case while for the magnetometer case the MSE does not spike until 1 s. The ideal vector case estimates become much better than the magnetometer case estimates after 1 s. Since control effort is most effective before the fins open at 1 s, the inaccuracy of the early estimates in the ideal case detrimentally affects the CEP.

We show in Figures 6.23 and 6.24 the normalized CEP resulting from the EKF and UKF estimates, respectively. The blue lines depict the baseline case, the green lines depict the controlled case, and the red lines depict the differences between the baseline and controlled cases. The EKF estimates improve the CEP from the baseline case by almost 9 units while the UKF only improves the baseline case by not quite 4 units.

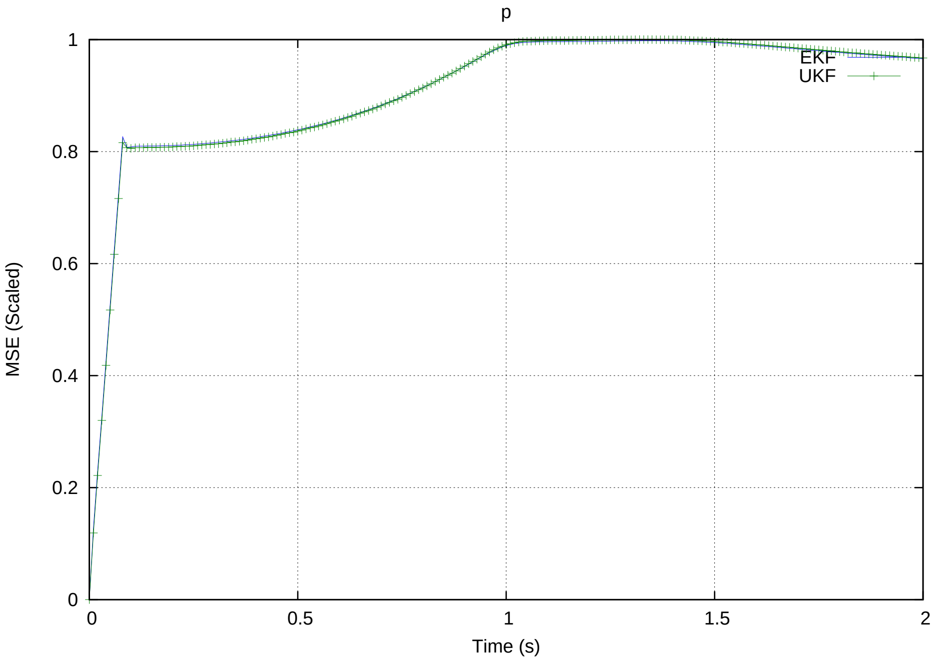

Figure 6.17: Scaled MSE of Roll Rates of EKF and UKF Estimates from Rate Gyros and a Magnetometer

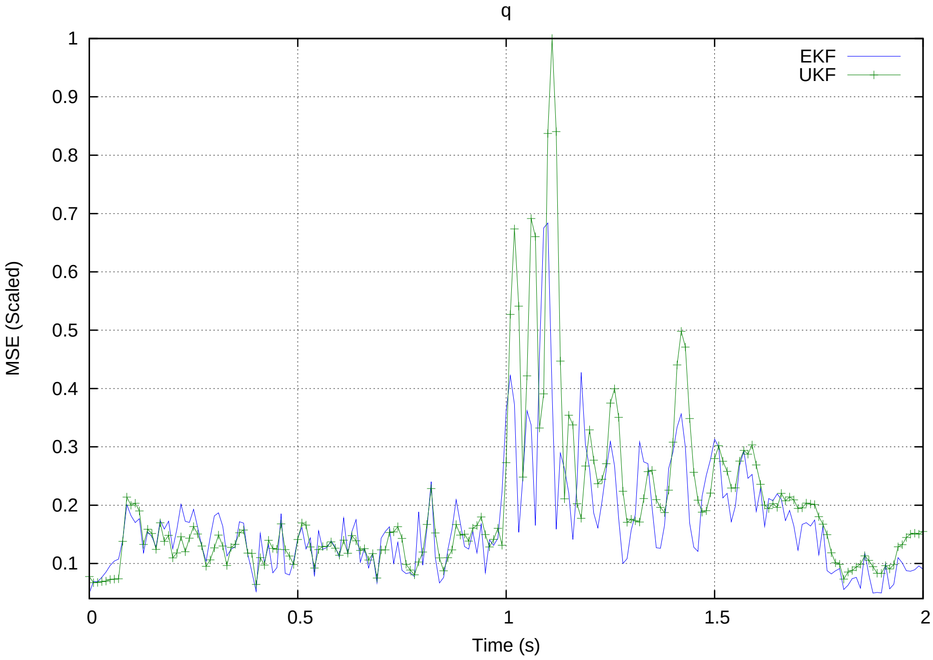

Figure 6.18: Scaled MSE of Pitch Rates of EKF and UKF Estimates from Rate Gyros and a Magnetometer

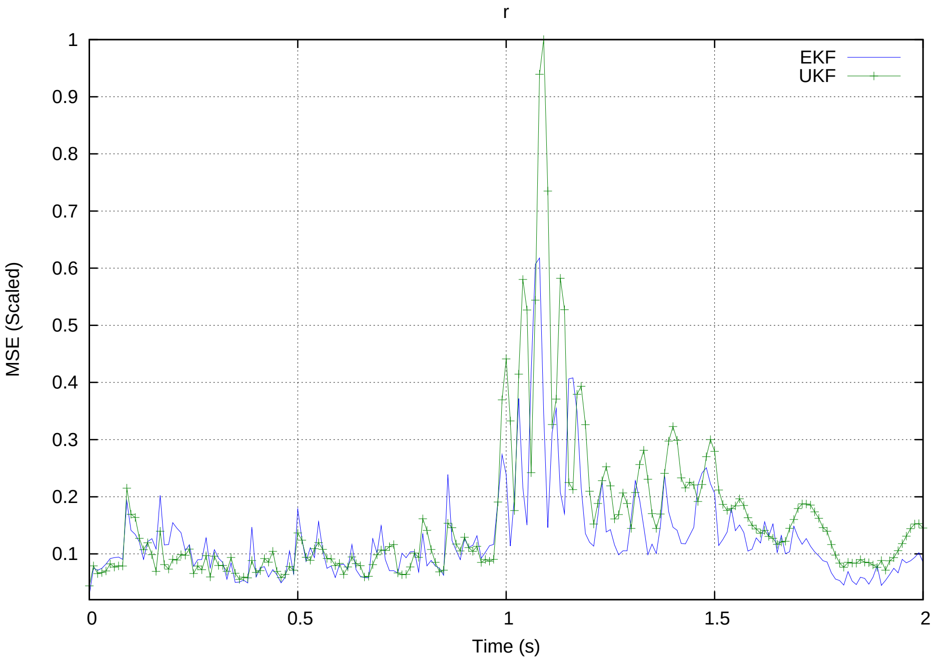

Figure 6.19: Scaled MSE of Yaw Rates of EKF and UKF Estimates from Rate Gyros and a Magnetometer

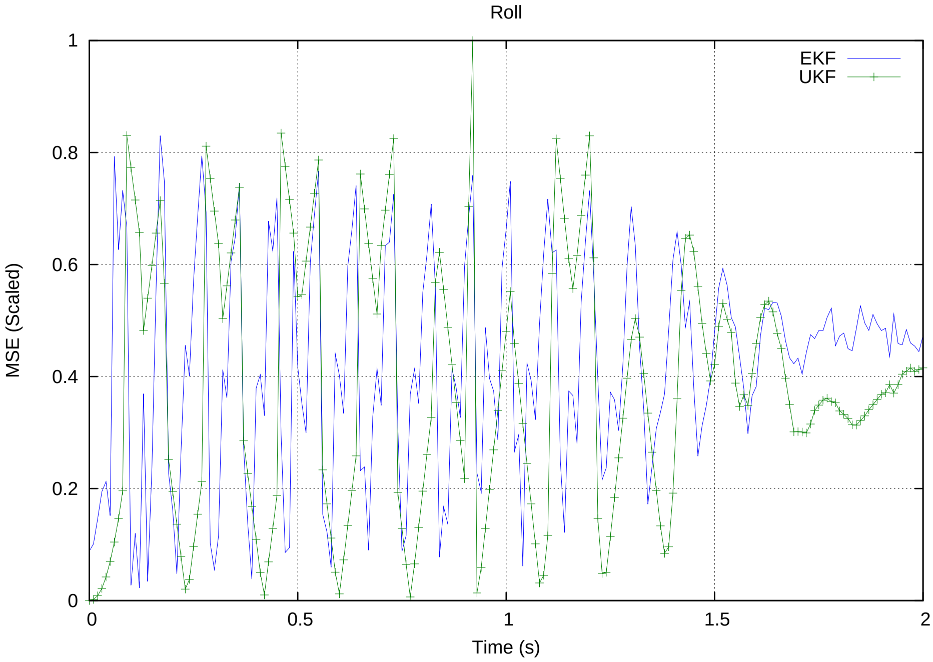

Figure 6.20: Scaled MSE of Roll Angles of EKF and UKF Estimates from Rate Gyros and a Magnetometer

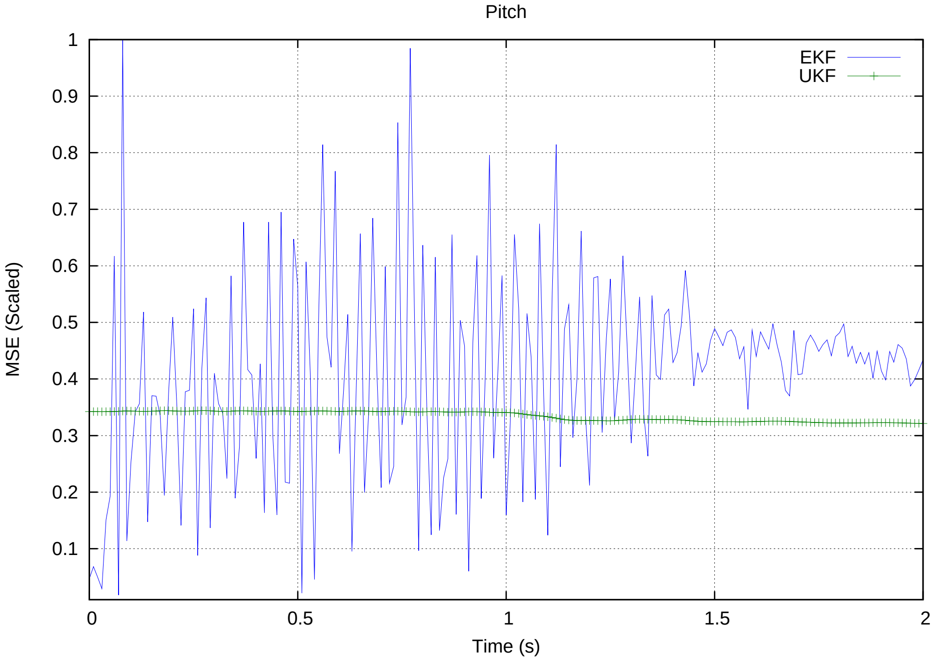

Figure 6.21: Scaled MSE of Pitch Angles of EKF and UKF Estimates from Rate Gyros and a Magnetometer

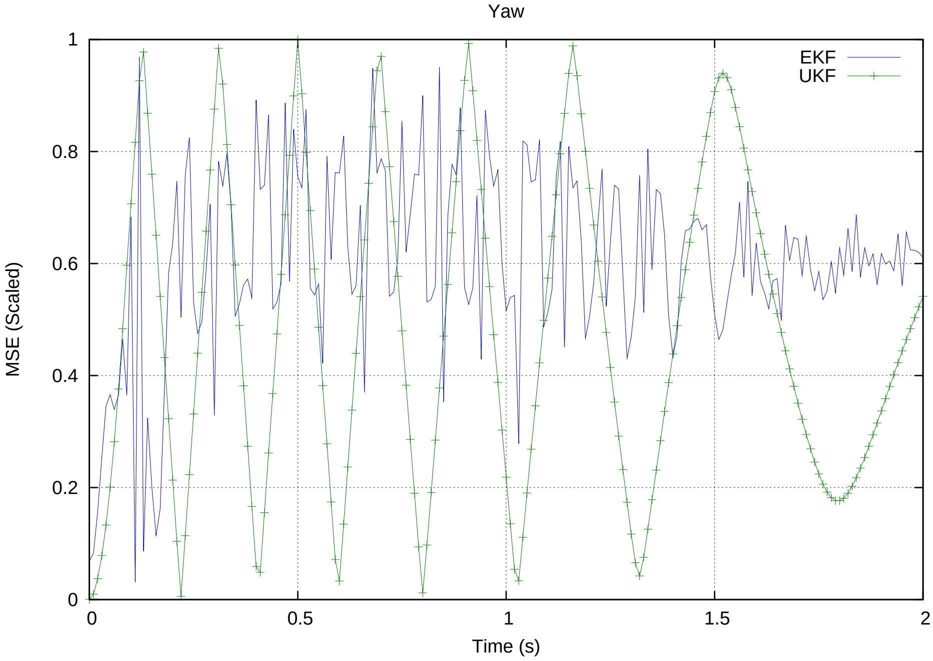

Figure 6.22: Scaled MSE of Yaw Angles of EKF and UKF Estimates from Rate Gyros and a Magnetometer

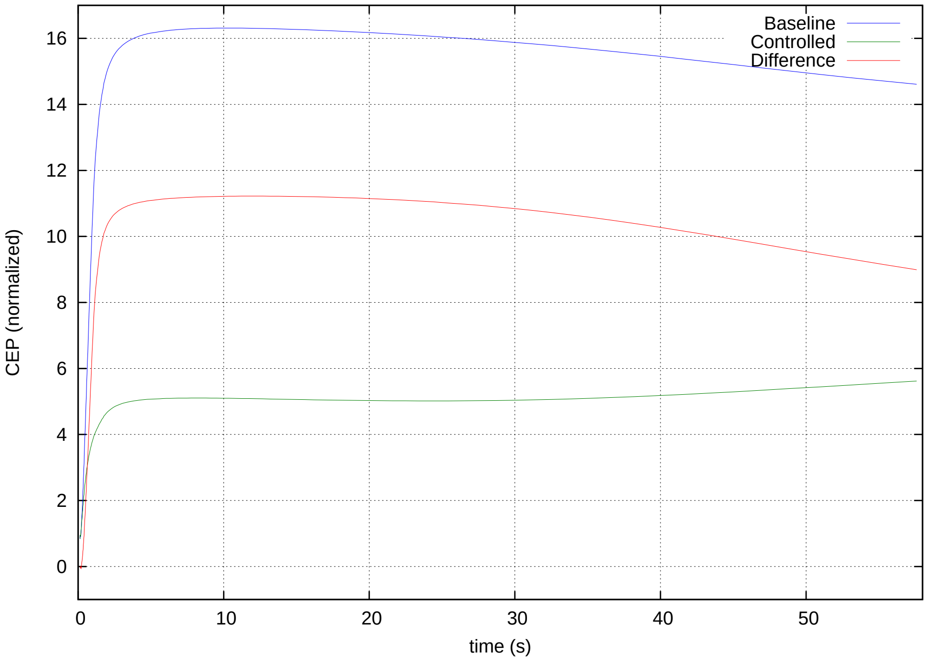

Figure 6.23: Normalized CEP for EKF Estimates from Rate Gyros and a Magnetometer

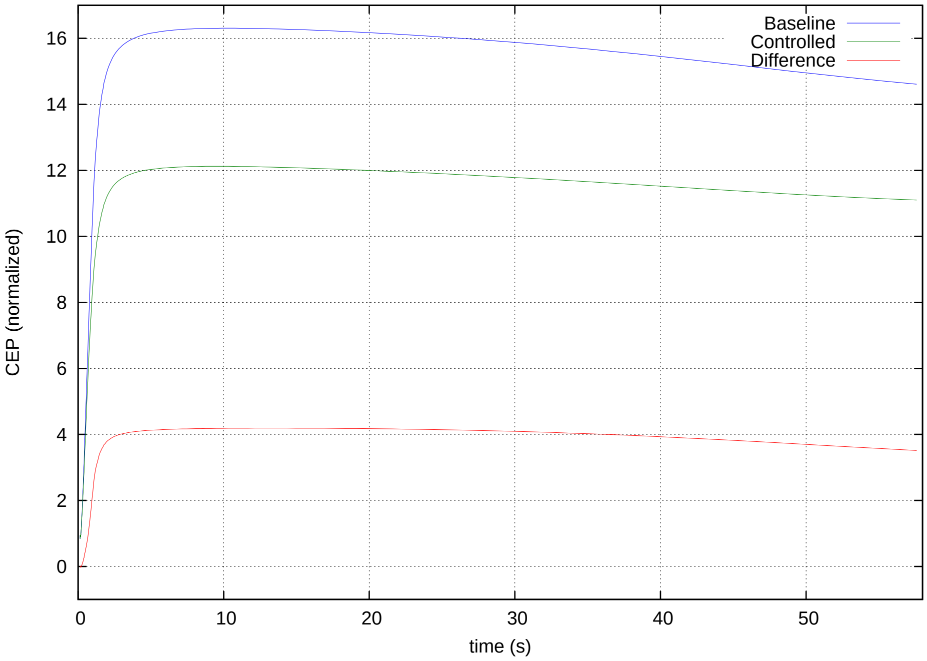

Figure 6.24: Normalized CEP for UKF Estimates from Rate Gyros and a Magnetometer

6.4 Rate Gyros and Angle Gyros

In this section we compare the EKF (implemented with Algorithm C.2.1 for the system model given by Equations (5.45) and (5.46) using the equations developed in §5.4.1) and UKF (implemented with Algorithm C.3.1 for the system model given by Equations (5.45) and (5.46)) performances with a sensor suite of both rate gyros and angle gyros. In Figures 6.25-6.27 and 6.28-6.30, we show the average MSE of EKF and UKF estimates of rotational rates and angular position, respectively. The EKF average MSEs are plotted in blue, and the UKF average MSEs are plotted in green. As we see from Figure 6.25, the EKF MSE is slightly better than the UKF MSE until shortly after 1.5 s, when the UKF MSE peaks while the EKF MSE begins to decrease. In Figures 6.26 and 6.27 we see that the EKF and UKF MSEs are almost identical, with the EKF MSE being only slightly better. Both UKF pitch rate and yaw rate MSEs spike to 1 shortly after launch while the EKF MSEs remain constant.

In Figures 6.28, 6.29, and 6.30 we show the average MSE of the EKF and UKF estimates for the roll angle ϕ, pitch angle θ, and yaw angle ψ, respectively. The EKF average MSE for roll angle oscillates about 0.4. The UKF average MSE oscillates about 0.5, and the error spikes to values greater than or equivalent to that of the EKF error. For pitch angle the UKF average MSE remains constant at a value slightly greater than 0.2. The UKF average MSE remains smaller than the EKF average MSE, which increases to 1 before 0.25 s and then settles to 0.8. For yaw angle the EKF average MSE settles to slightly greater than 0.4 after spiking to 0.7 before 0.25 s. The UKF average MSE oscillates between 0 and 1. After 1 s the magnitude of the oscillations begin to decrease.

In Figures 6.31 and 6.32 we show the normalized CEP. We show the CEP for EKF estimates in Figure 6.31 and the CEP for UKF estimates in Figure 6.32. The blue lines depict the baseline case, the green lines depict the controlled case, and the red lines depict the differences between the baseline and controlled cases. The EKF estimates reduce the controlled CEP from the baseline CEP by approximately 9 units while the UKF only reduces the CEP between the controlled and baseline cases by about 3 units.

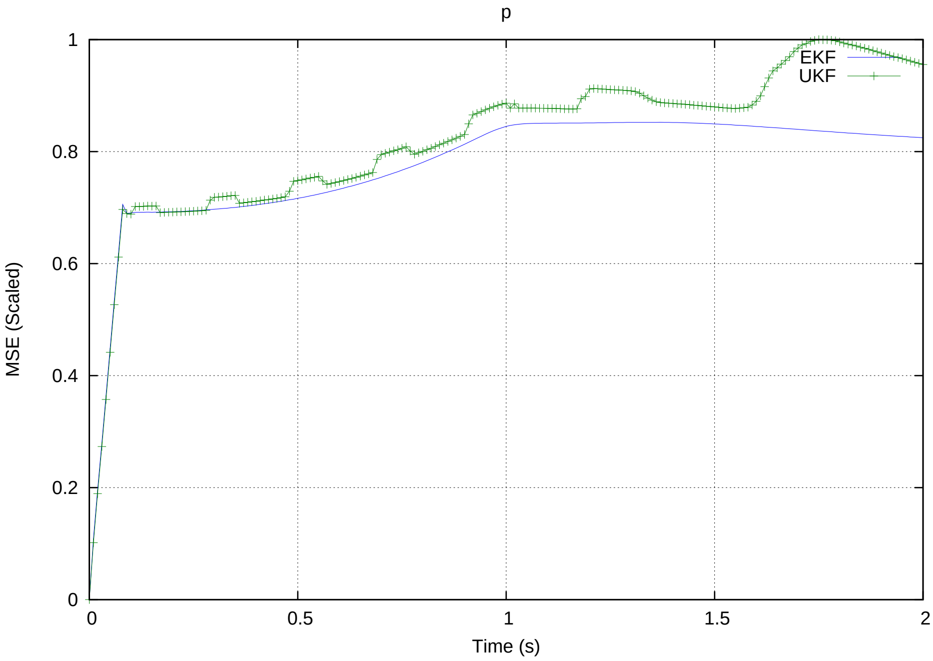

Figure 6.25: Scaled MSE of Roll Rates of EKF and UKF Estimates from Rate Gyros and Angle Gyros

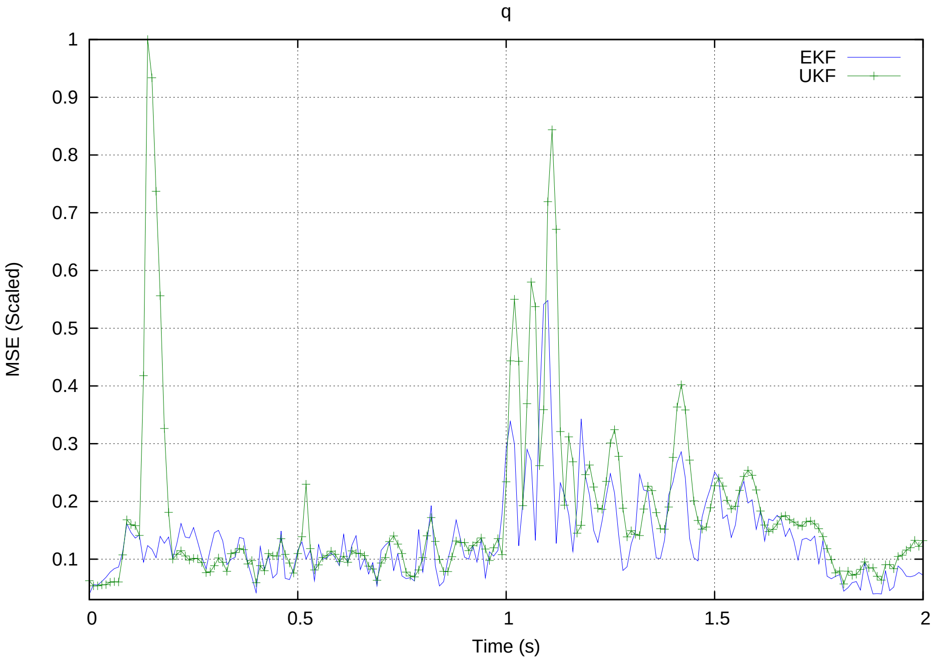

Figure 6.26: Scaled MSE of Pitch Rates of EKF and UKF Estimates from Rate Gyros and Angle Gyros

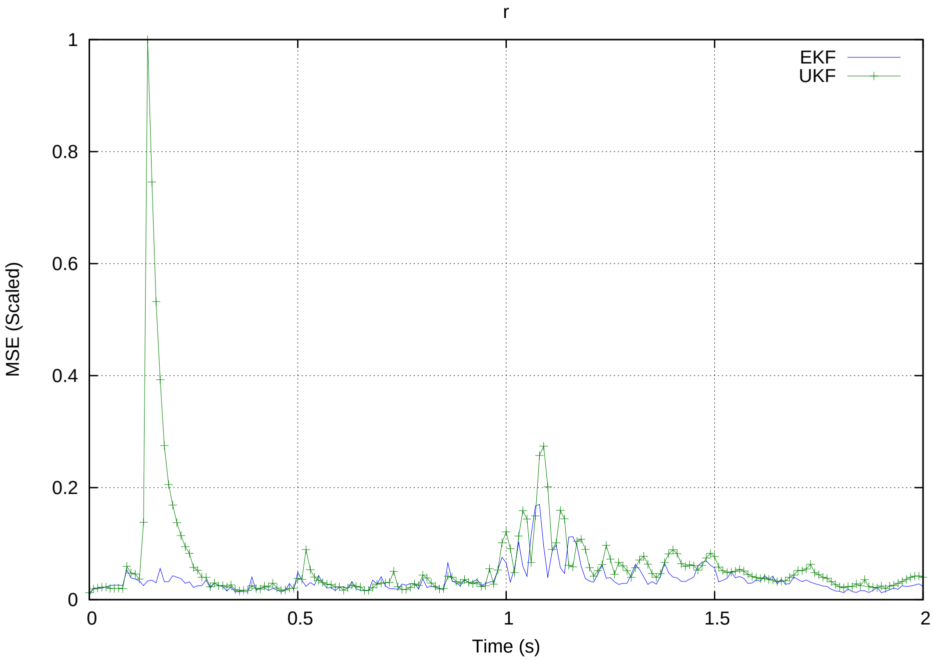

Figure 6.27: Scaled MSE of Yaw Rates of EKF and UKF Estimates from Rate Gyros and Angle Gyros

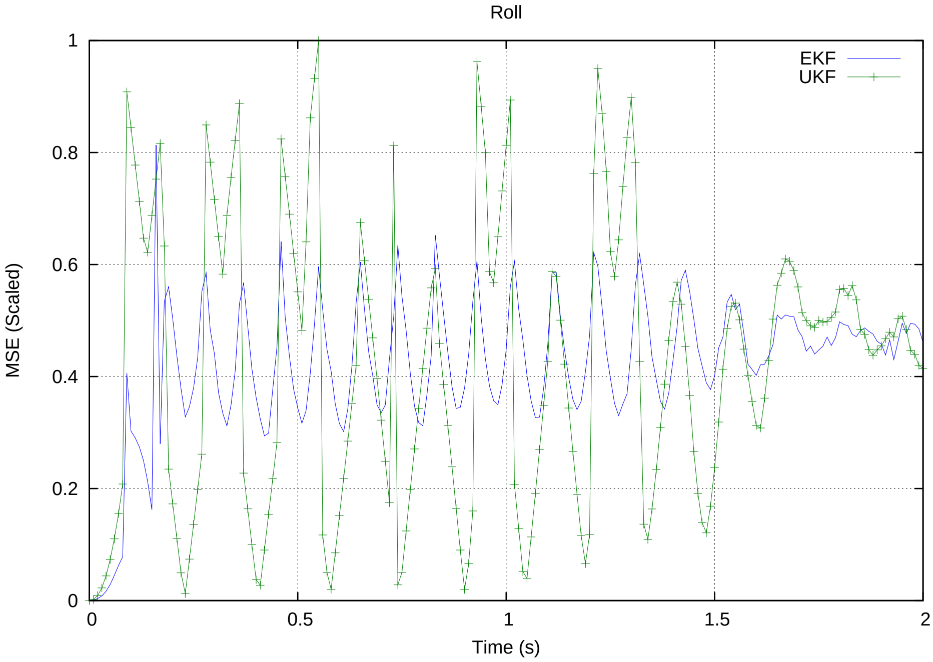

Figure 6.28: Scaled MSE of Roll Angles of EKF and UKF Estimates from Rate Gyros and Angle Gyros

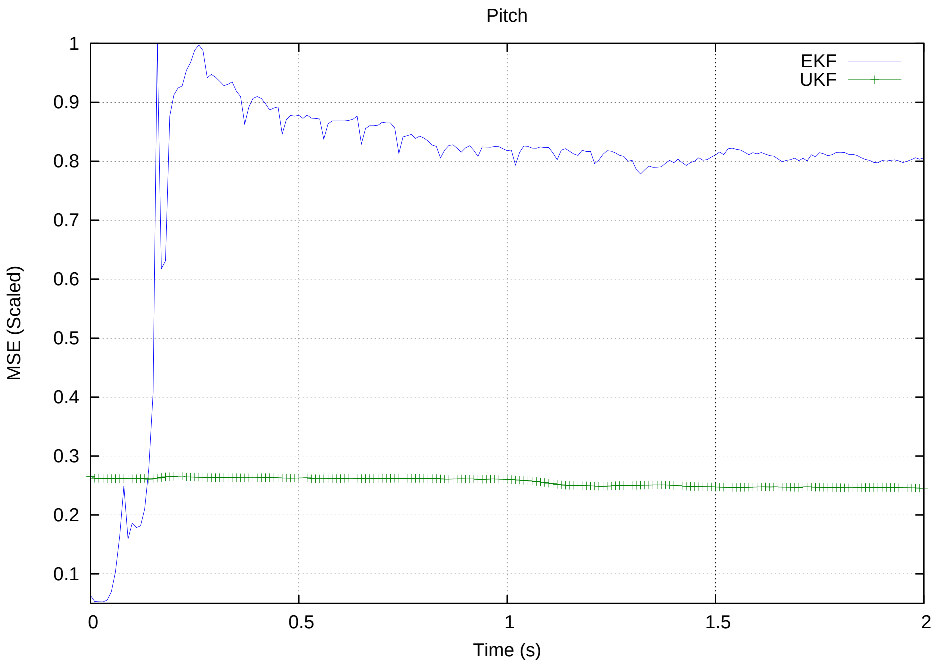

Figure 6.29: Scaled MSE of Pitch Angles of EKF and UKF Estimates from Rate Gyros and Angle Gyros

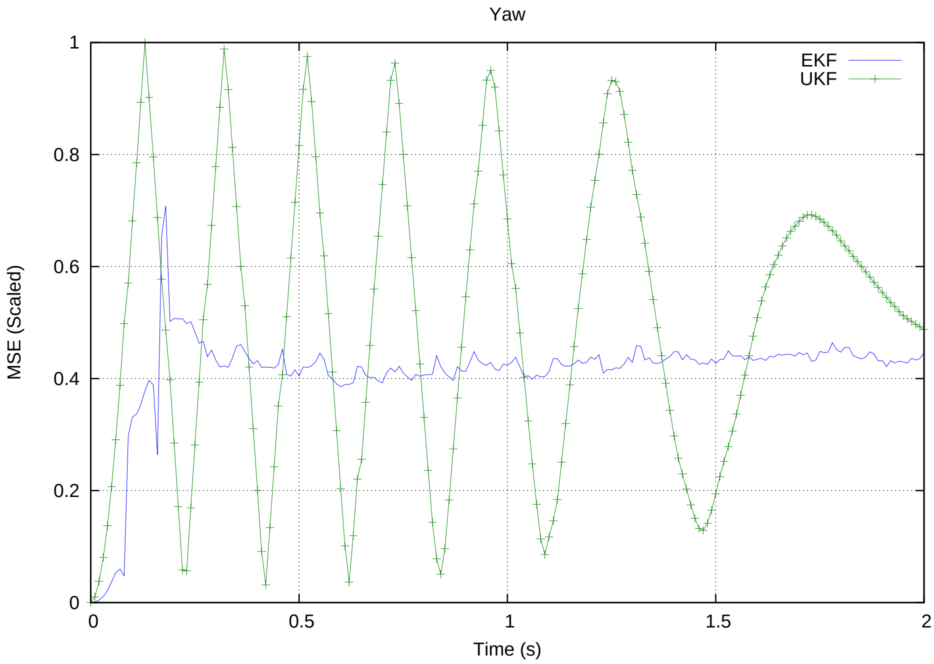

Figure 6.30: Scaled MSE of Yaw Angles of EKF and UKF Estimates from Rate Gyros and Angle Gyros

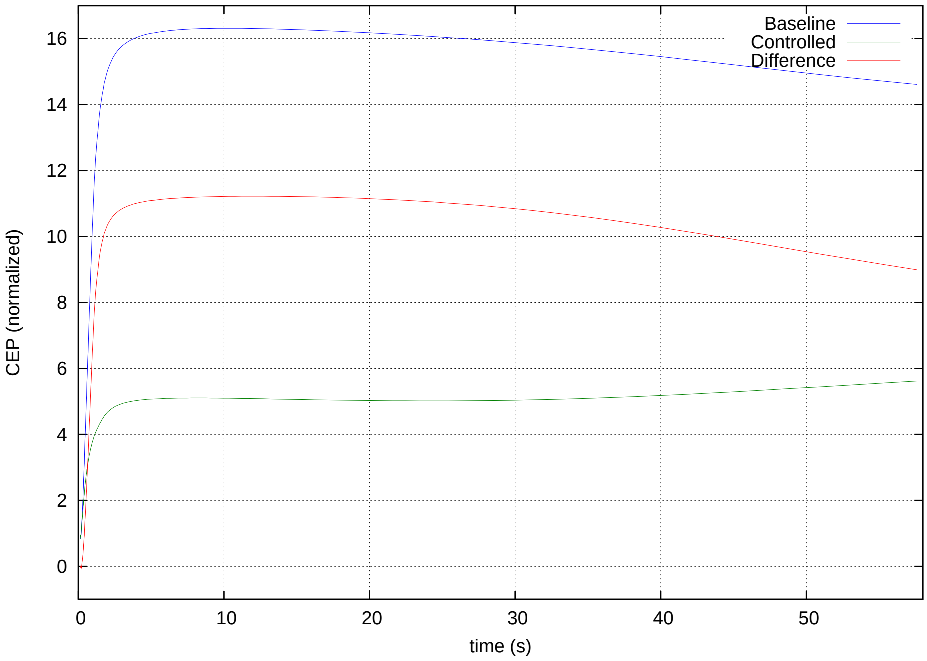

Figure 6.31: Normalized CEP for EKF Estimates from Rate Gyros and Angle Gyros

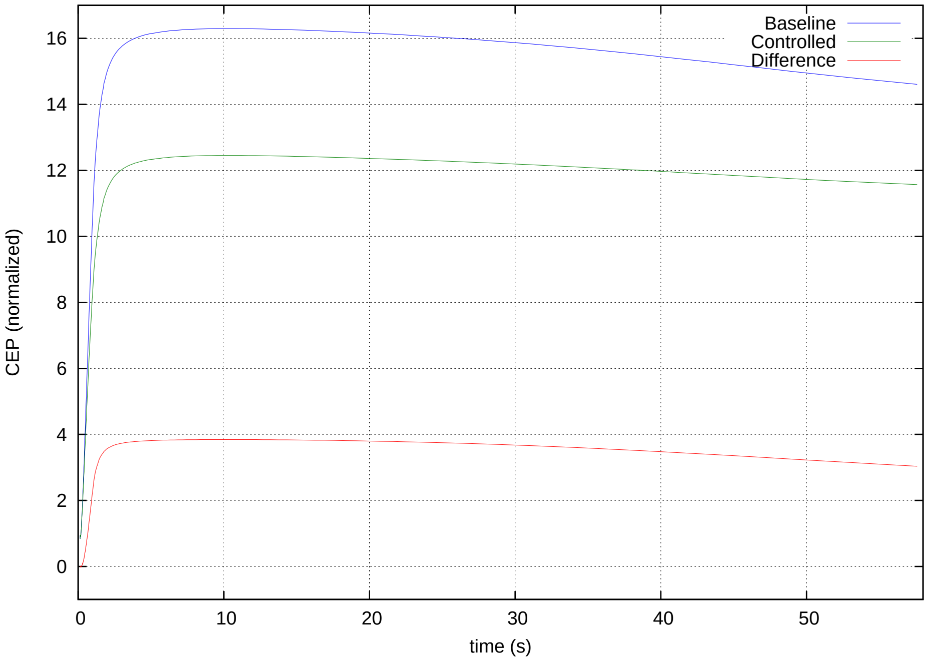

Figure 6.32: Normalized CEP for UKF Estimates from Rate Gyros and Angle Gyros

6.5 Rate Gyros, Angle Gyros, and Magnetometer

In this section we focus on the results of using the ESOQ algorithm to estimate the rocket's states. In §6.5.1 we present the results of ESOQ state estimations with ideal sensors and nonideal sensors. Next, in §6.5.2 we present the results of using the EKF and UKF to estimate states when rate gyro, angle gyro, and magnetometer measurements are all available in order to compare the results to ESOQ-based estimators in following sections. Finally, in §6.5.3 we present the results of combining the ESOQ algorithm with the KF, EKF, and UKF.

6.5.1 ESOQ Algorithm

In Figures 6.33 and 6.34 we show the results of the rocket being controlled with estimates from the ESOQ algorithm. The only sensors used for these simulations are angle gyros and a magnetometer. In Figure 6.33 the results of the ESOQ algorithm are shown when ideal sensors are used. The top line indicates the baseline CEP when no control is applied to the rocket, the middle line is the rocket's controlled CEP, and the bottom line is the difference between the two cases. The ESOQ estimator with ideal sensor readings alone only improves the the CEP by about 1.5 units. The CEP resulting from nonideal sensor readings is shown in Figure 6.34. The top line is the controlled CEP, the middle line is the baseline CEP, and the bottom line is the difference between the two CEPs. The controlled CEP is worse than the baseline CEP when nonideal sensor readings are provided to the ESOQ algorithm. One reason for these results is that the control law relies heavily on rotational rate estimates. The ESOQ algorithm only provides angular position estimates so rotational rates are estimated by Euler differentiation. We show normalized roll rates for all 250 dispersion runs in Figure 6.35, and we show the normalized roll rates as estimated by the ESOQ algorithm with nonideal sensors in Figure 6.36. In Figure 6.37 we show the normalized actual roll angles, and in Figure 6.38 we show the normalized roll angles as estimated by the ESOQ algorithm with nonideal sensors. While the ESOQ algorithm predicts roll angle well, the Euler differentiation yields poor estimates of roll rate. The ESOQ algorithm yields similar results for pitch angles and rates as shown in Figures 6.39-6.42. As we see from Figures 6.43-6.46, the ESOQ algorithm poorly estimates yaw angles and therefore yaw rates.

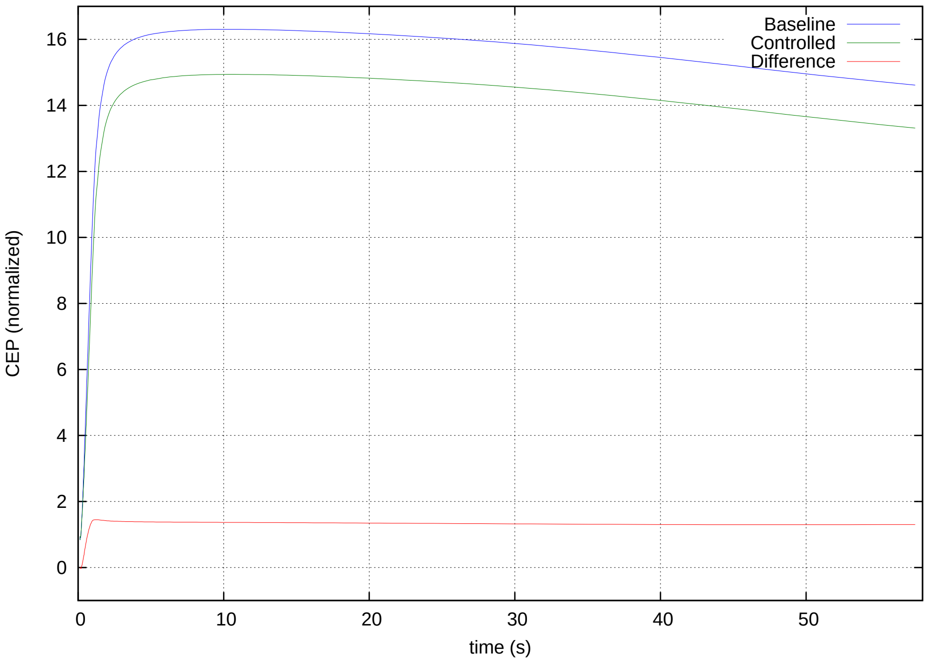

Figure 6.33: CEP for ESOQ Controlled Rocket with Ideal Angle Gyros and Magnetometer

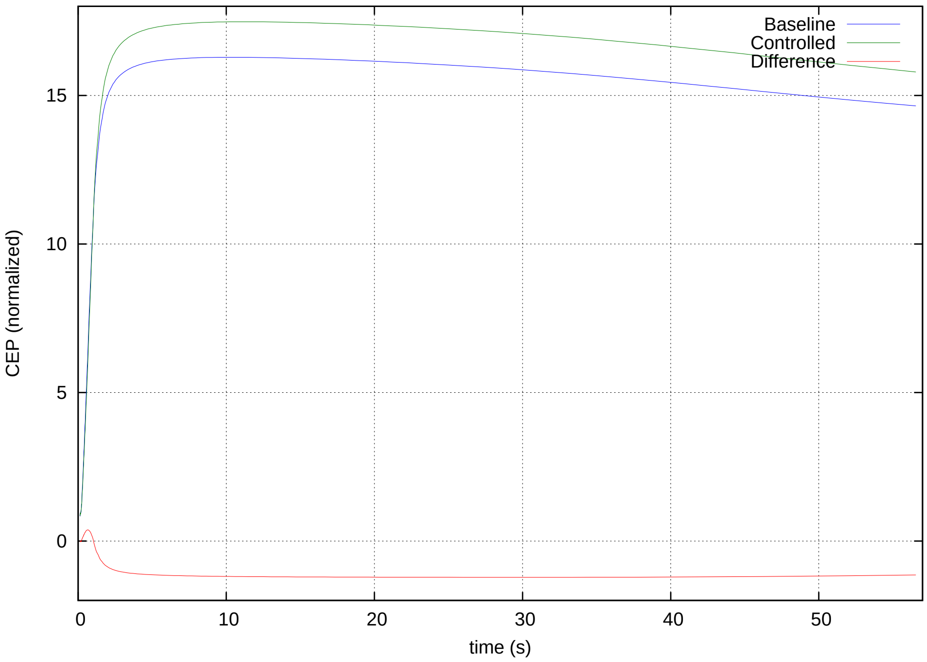

Figure 6.34: CEP for ESOQ Controlled Rocket with Nonideal Angle Gyros and Magnetometer

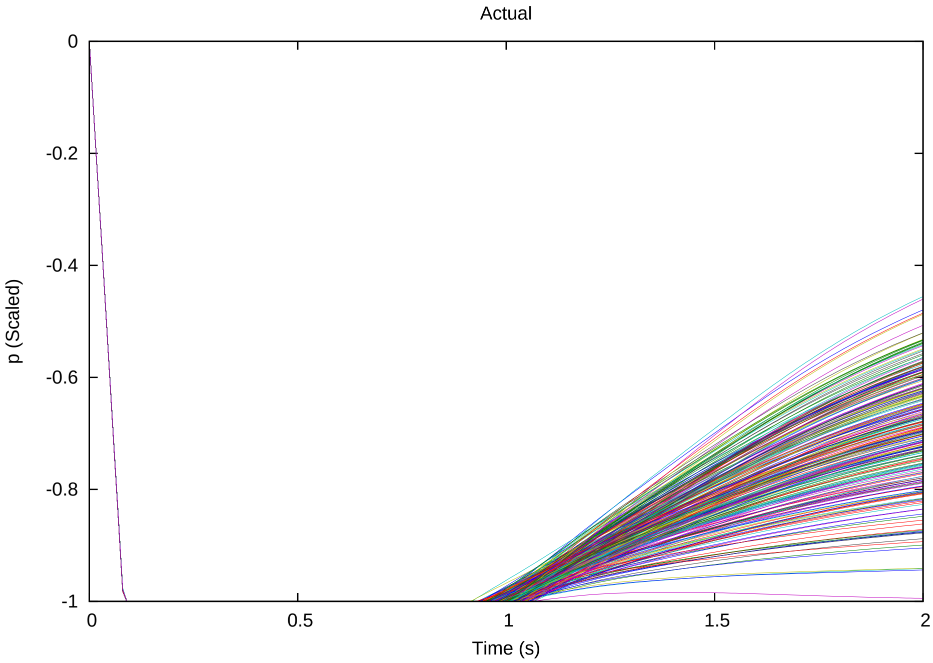

Figure 6.35: Actual Roll Rates (Normalized)

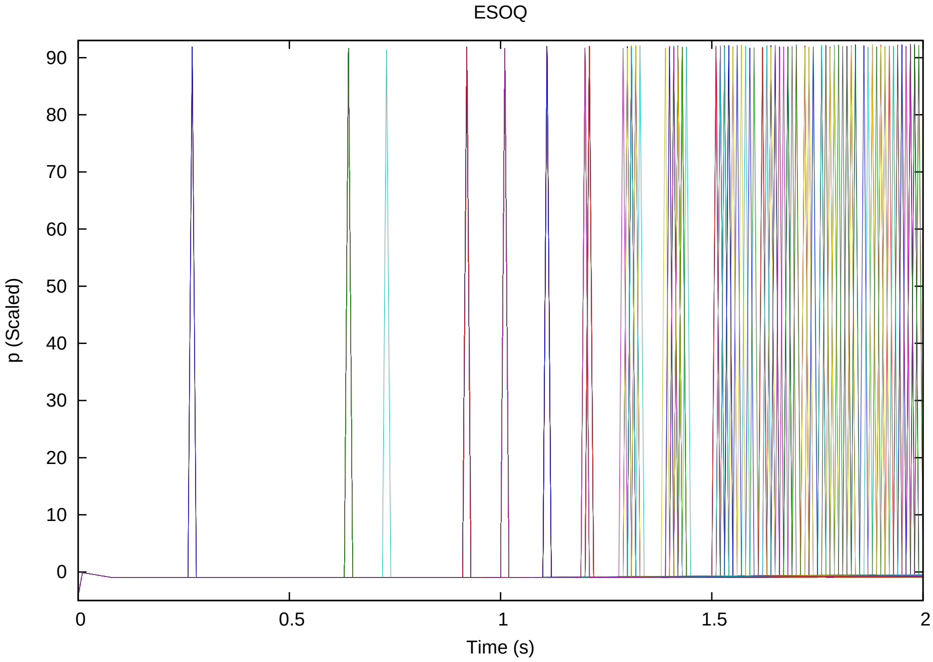

Figure 6.36: ESOQ Estimated Roll Rates with Nonideal Sensors (Normalized)

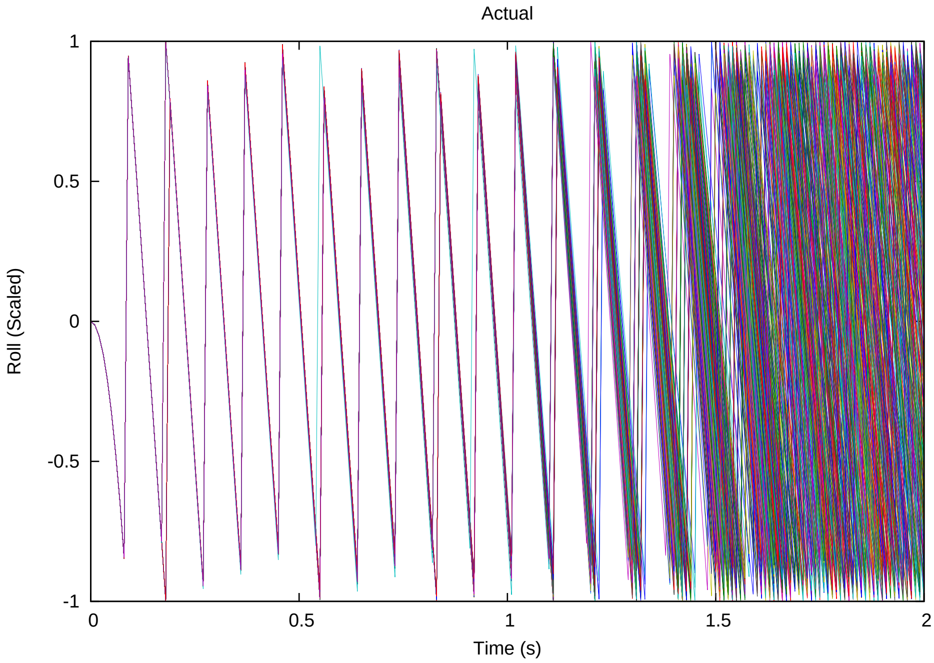

Figure 6.37: Actual Roll Angles (Normalized)

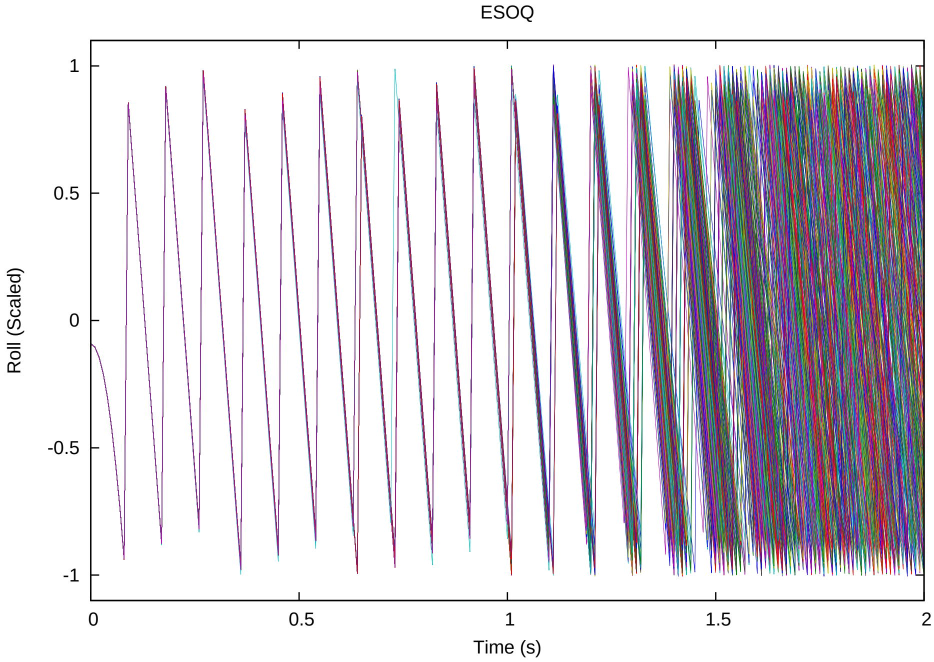

Figure 6.38: ESOQ Estimated Roll Angles with Nonideal Sensors (Normalized)

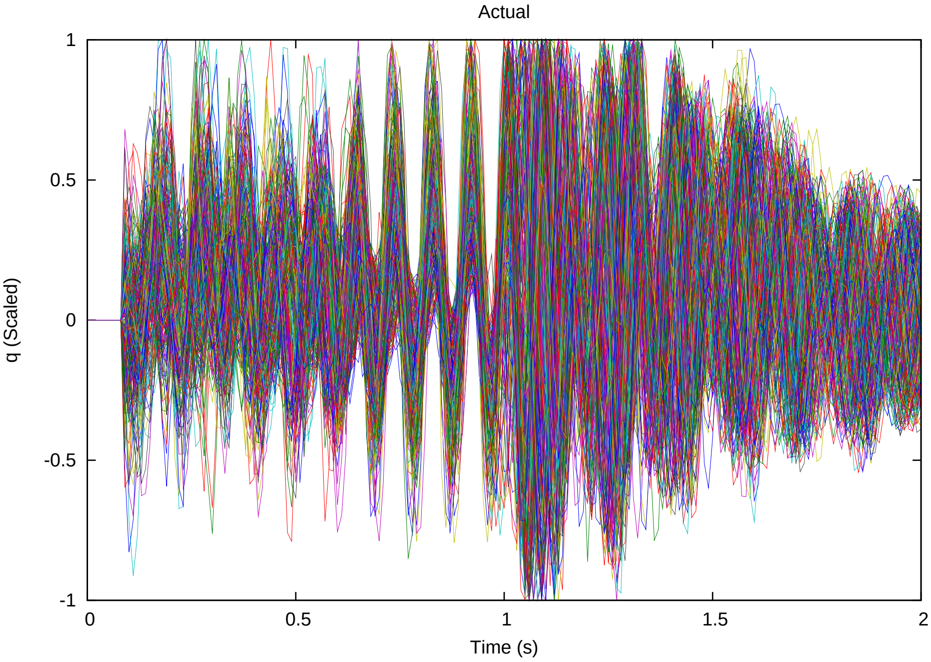

Figure 6.39: Actual Pitch Rates (Normalized)

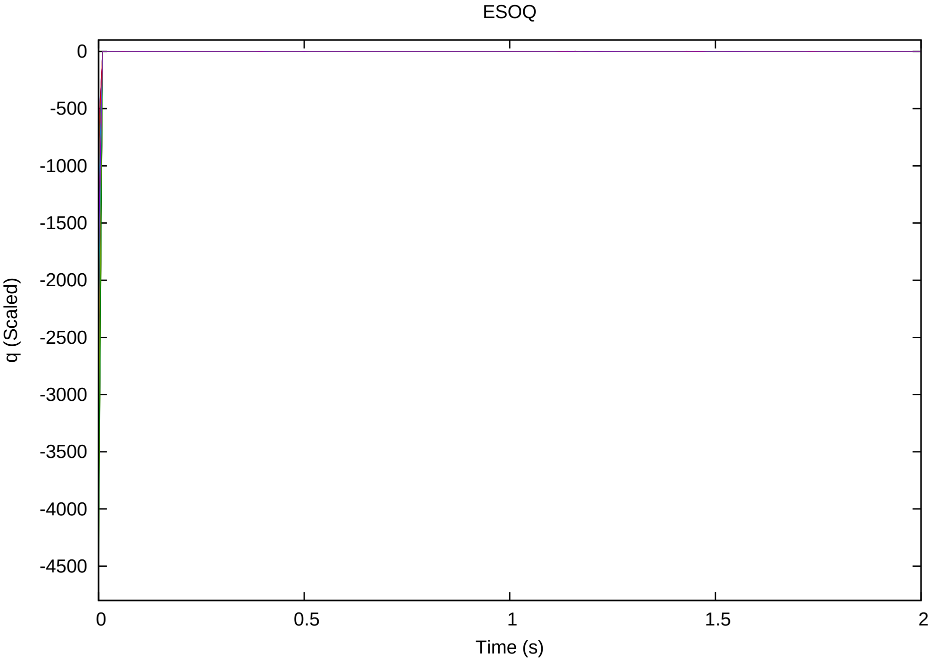

Figure 6.40: ESOQ Estimated Pitch Rates with Nonideal Sensors (Normalized)

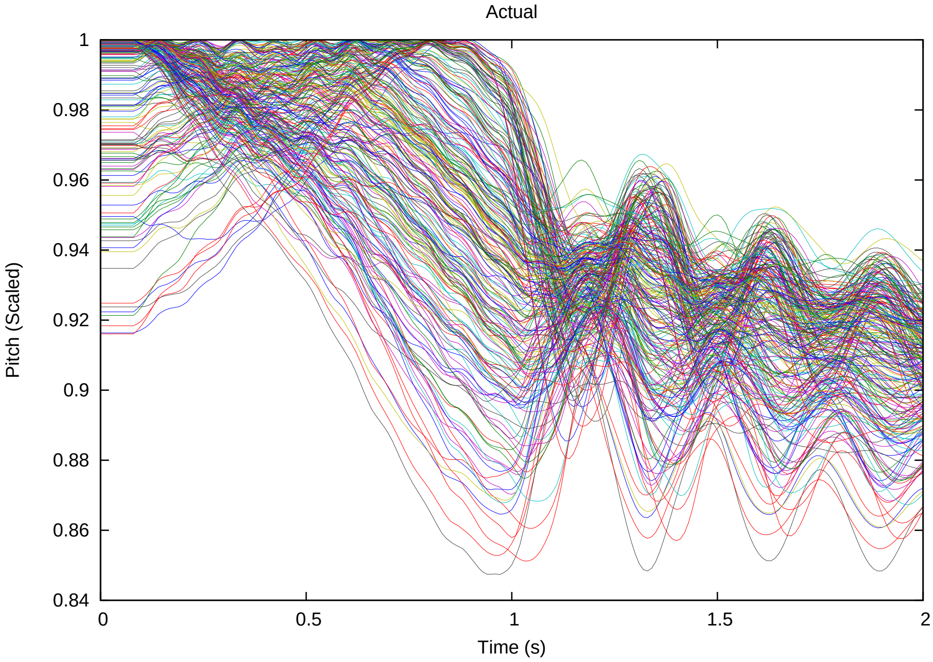

Figure 6.41: Actual Pitch Angles (Normalized)

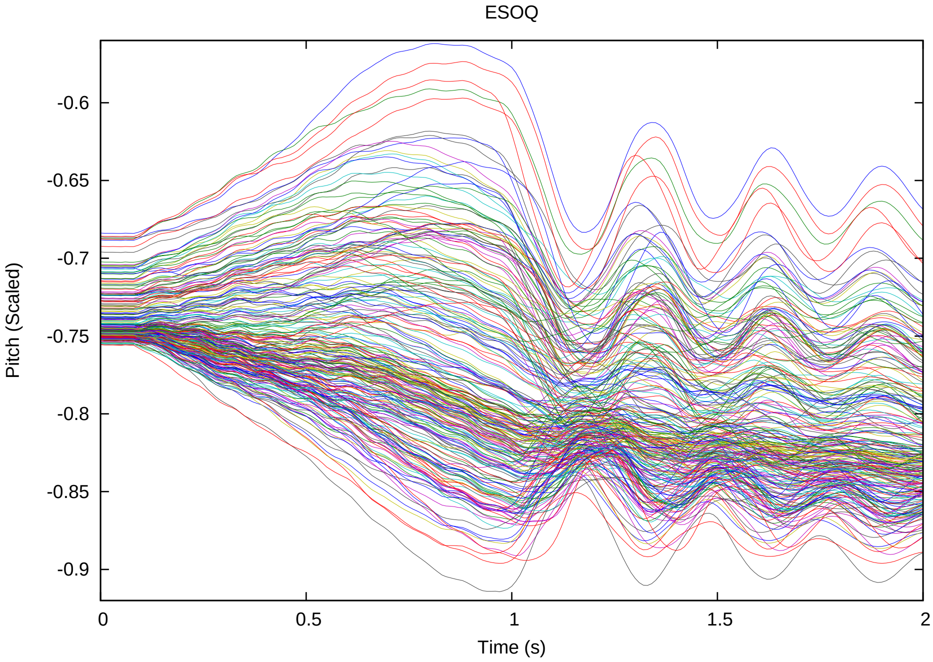

Figure 6.42: ESOQ Estimated Pitch Angles with Nonideal Sensors (Normalized)

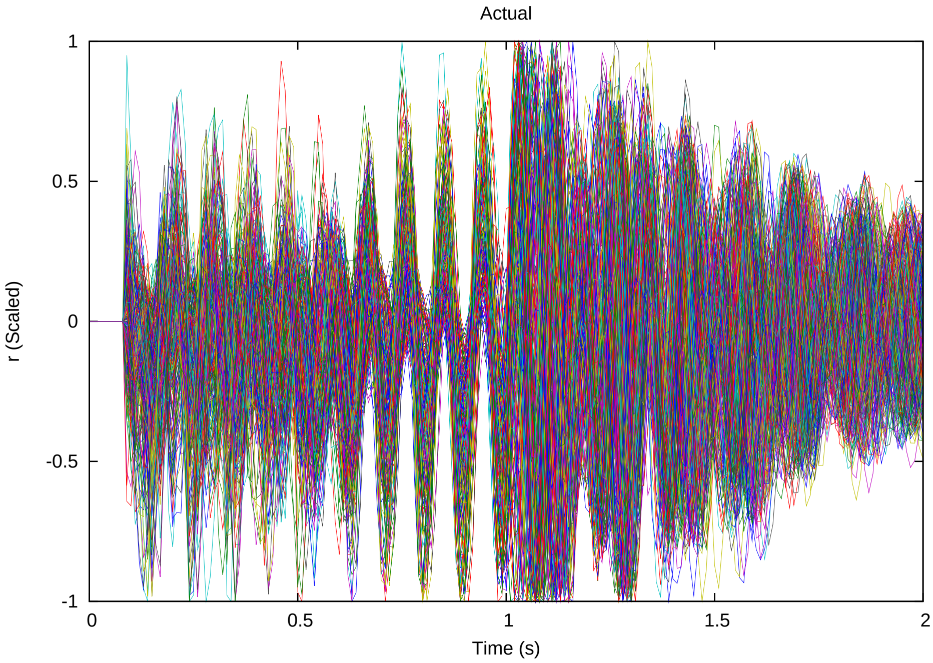

Figure 6.43: Actual Yaw Rates (Normalized)

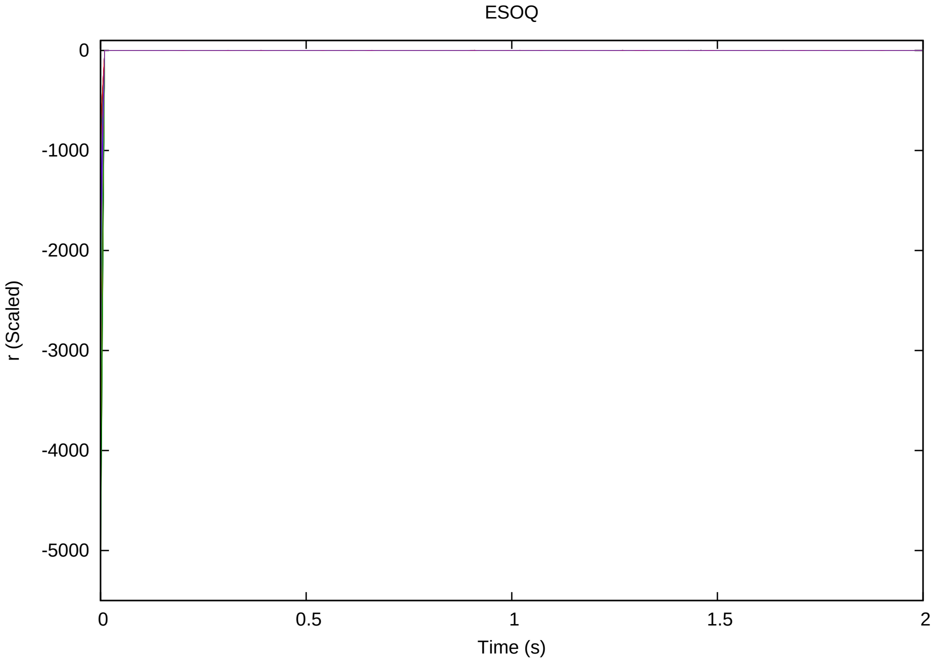

Figure 6.44: ESOQ Estimated Yaw Rates with Nonideal Sensors (Normalized)

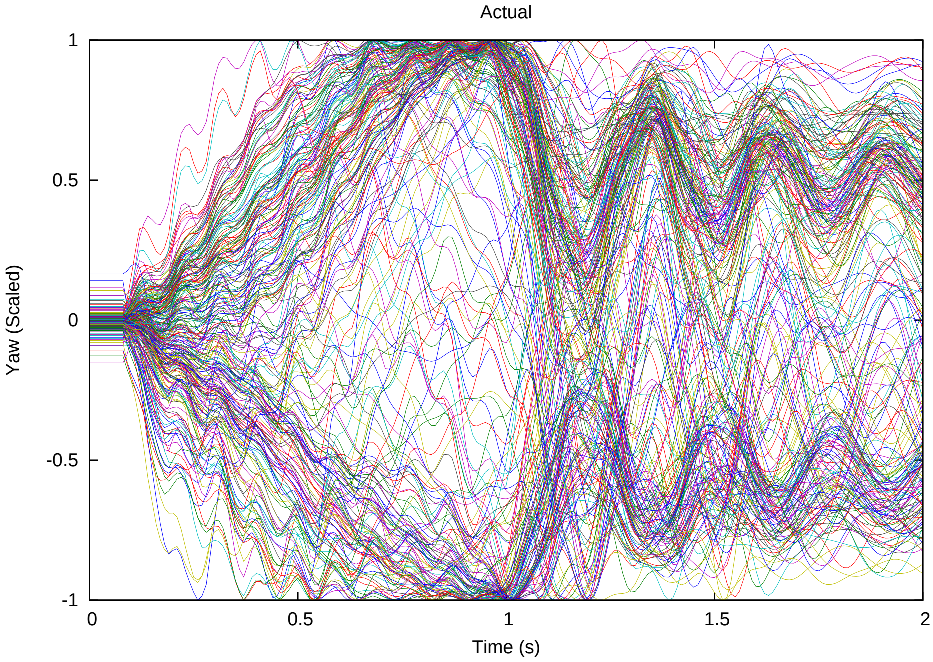

Figure 6.45: Actual Yaw Angles (Normalized)

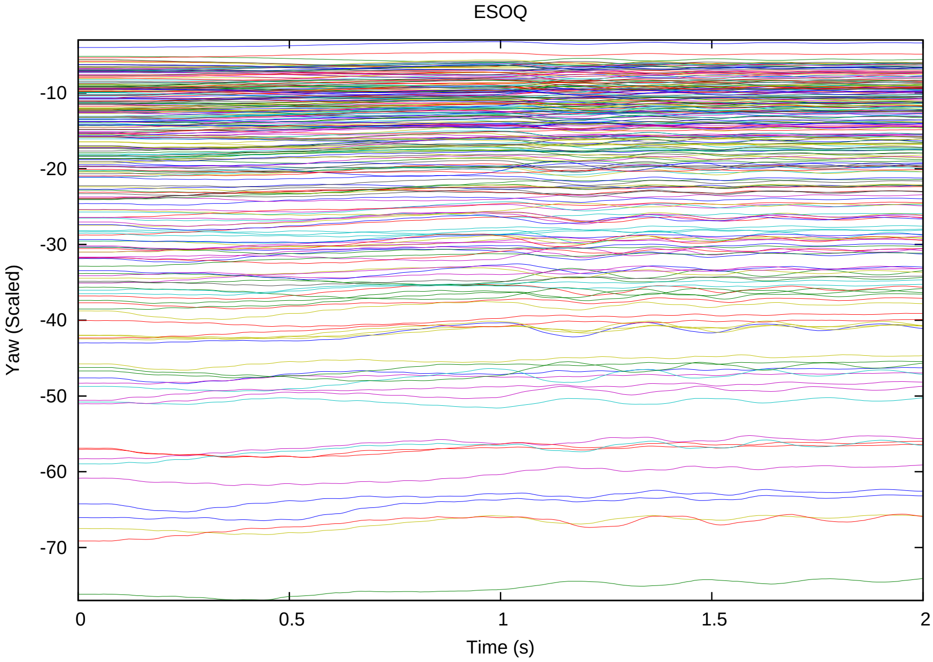

Figure 6.46: ESOQ Estimated Yaw Angles with Nonideal Sensors (Normalized)

6.5.2 EKF and UKF with All Sensors

We show the CEPs resulting from EKF and UKF estimates with rate gyros, angle gyros, and a magnetometer as described by Equations (5.62) and (5.63) in Figures 6.47 and 6.48, respectively. In Figure 6.47 the top line is the baseline CEP, the bottom line is the EKF-controlled CEP, and the middle line is the difference between the two. The EKF performs well reducing the CEP by about 9 units. The UKF, however, performs poorly as we show in Figure 6.48, where the top line is the UKF controlled CEP, the middle line is the baseline CEP, and the bottom line is the difference between the baseline and controlled cases. The CEP is actually increased over the baseline case by about 1 unit.

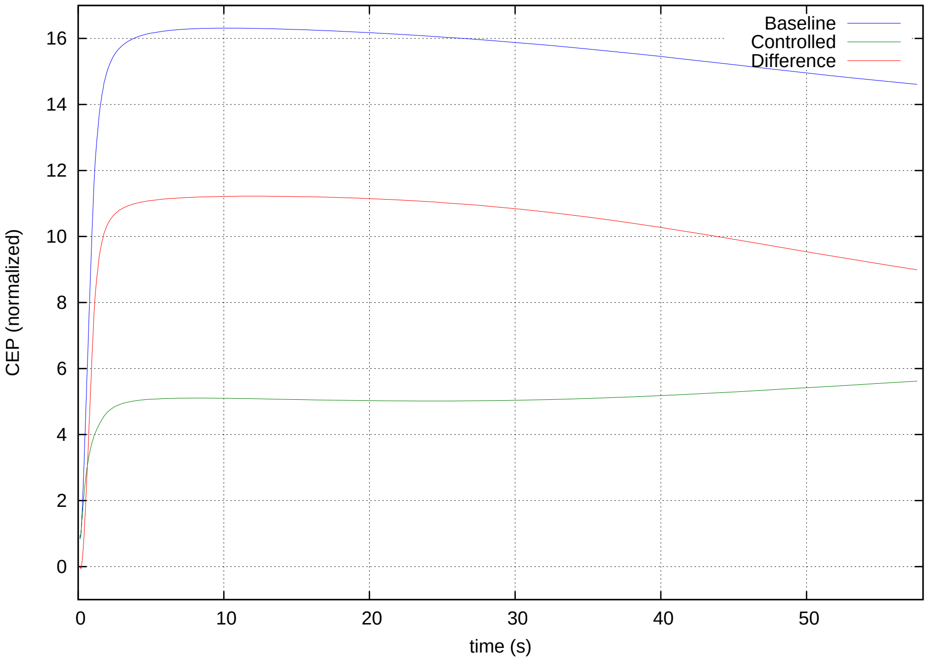

Figure 6.47: CEP for EKF Controlled Rocket with Rate Gyros, Angle Gyros, and Magnetometer

6.5.3 ESOQ Algorithm Combined with Kalman Filters

In Figures 6.49-6.52 we show the CEPs resulting from combining estimates from the ESOQ algorithm with Kalman filter based estimators. We show the results of attitude estimates from the ESOQ algorithm being used as inputs to an EKF as described by Equations (5.64) and (5.65) in

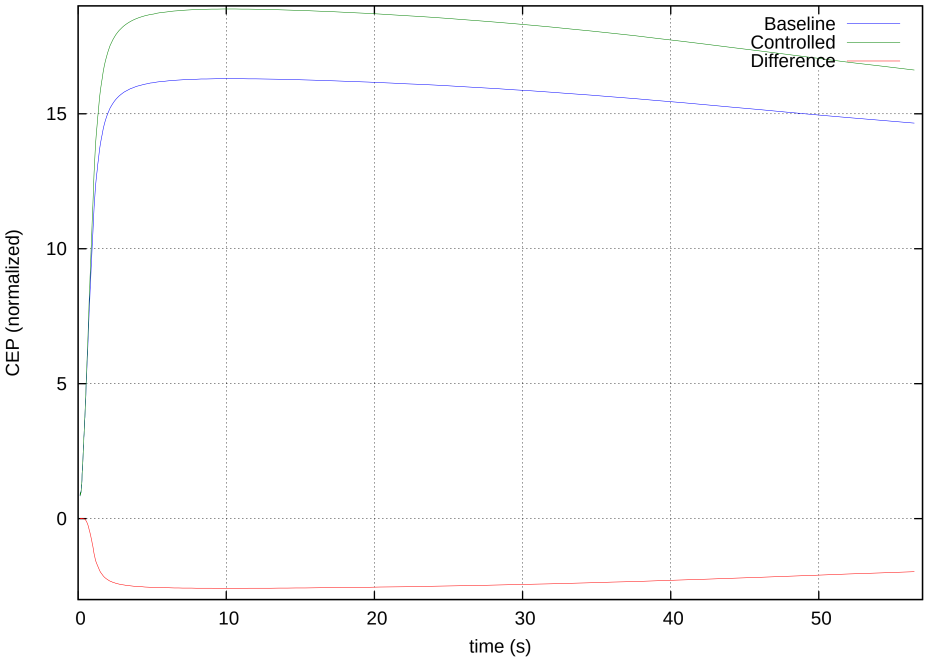

Figure 6.48: CEP for UKF Controlled Rocket with Rate Gyros, Angle Gyros, and Magnetometer

Figure 6.49. For this case the ESOQ algorithm provides an estimate of the rocket's attitude in the form of a quaternion. This attitude estimate is then treated as a sensor reading which is provided to the EKF. The EKF then provides rotational rate and attitude estimates based on readings from the rate gyros and the ESOQ algorithm estimates. In Figure 6.49 the top line is the baseline CEP, the bottom line is the controlled CEP, and the middle line is the difference between the two cases. The CEP is improved by about 9 units. In Figure 6.50 the results of controlling the rocket with estimates from a EKF/ESOQ hybrid are shown. The EKF is used to estimate angular rates, and the ESOQ algorithm is used to estimate angular position. The top line in the figure is the baseline CEP, the bottom line is the controlled CEP, and the middle line is the difference between them. The CEP is improved by about 9 units. We show the results for a hybrid UKF/ESOQ estimator in Figure 6.51. The top line in Figure 6.51 is the baseline CEP, the middle line is the controlled CEP, and the bottom line is the difference between the baseline and controlled cases. The UKF/ESOQ hybrid estimator improves the CEP by just under 4 units. In Figure 6.52 is shown the results of the rocket controlled with a hybrid KF/ESOQ estimator, where the KF estimates rotational rates, and the ESOQ algorithm estimates attitude. The top line in the figure is the baseline CEP, the

bottom line is the controlled CEP, and the middle line is their difference. The KF/ESOQ estimator improves the CEP by slightly more than 9 units.

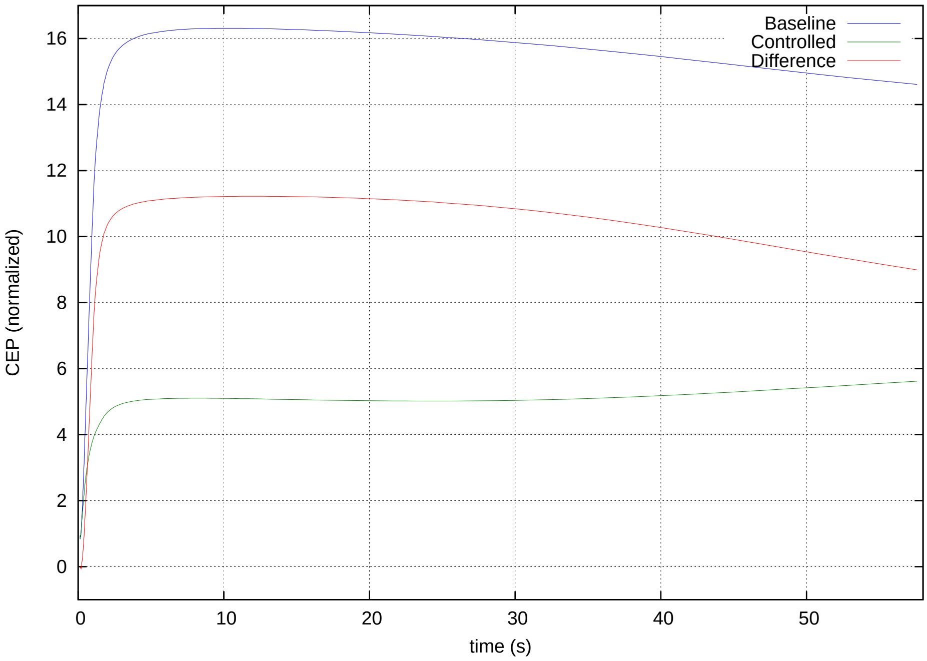

Figure 6.49: CEP for Rocket Controlled by Estimates from an EKF with Inputs from the ESOQ Algorithm

In Figures 6.53-6.58 we show the MSE of the state estimates of the EKF, UKF, and the EKF that has ESOQ estimates as inputs. Roll rate, pitch rate, and yaw rate MSEs are shown in Figures 6.53, 6.54, and 6.55, respectively. In these three figures the EKF MSE and the EKF/ESOQ MSE are identical as shown by the blue EKF line being directly under the red EKF/ESOQ line. The EKF estimators have better MSEs than the UKF roll rate until the fins open at 1 s. At this point the UKF MSE steadily decreases. The UKF MSE for pitch rate is much smoother than the MSEs for the EKF estimator. The UKF MSE spikes shortly after the rocket's launch and then falls below the EKF MSEs after 0.2 s. The MSEs of the EKF estimator begin to spike when the rocket's fins open. For yaw rate the UKF MSE peaks shortly after 0.1 s at which point it slowly decreases, not falling below the MSEs of the EKF estimator until shortly before 1 s. As in the pitch rate case, the EKF estimators' MSEs spike when the rocket's fins open. In Figures 6.56, 6.57, and 6.58 we show the roll angle, pitch angle, and yaw angle MSEs, respectively. The roll angle MSEs for all

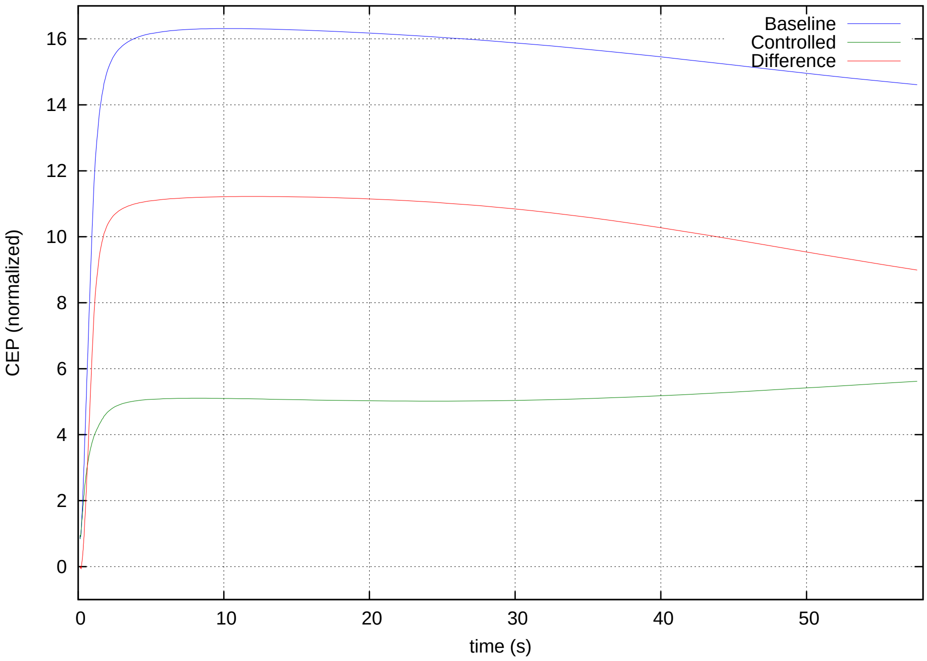

Figure 6.50: CEP for EKF/ESOQ Hybrid Estimator Controlled Rocket

three estimators oscillate with there being little difference between the EKF and UKF estimators' MSEs. The EKF/ESOQ estimator's MSE stays less than the MSEs of the other estimator after 1.5 s. For pitch angle the UKF yields the worst MSE, which consistently stays near 0.8 units. The MSEs of the EKF and EKF/ESOQ estimators oscillate with the EKF settling to a slightly higher value of 0.6 units than the EKF/ESOQ estimator, which settles to around 0.55 units. The yaw angle MSEs behave similarly to the pitch angle MSEs. The UKF MSE stays consistently around 0.65 units, while the MSEs of the other estimators oscillate. Unlike the pitch angle MSE, the EKF MSE settles to a lower value of 0.6 while the EKF/ESOQ MSE reaches about 0.7 units.

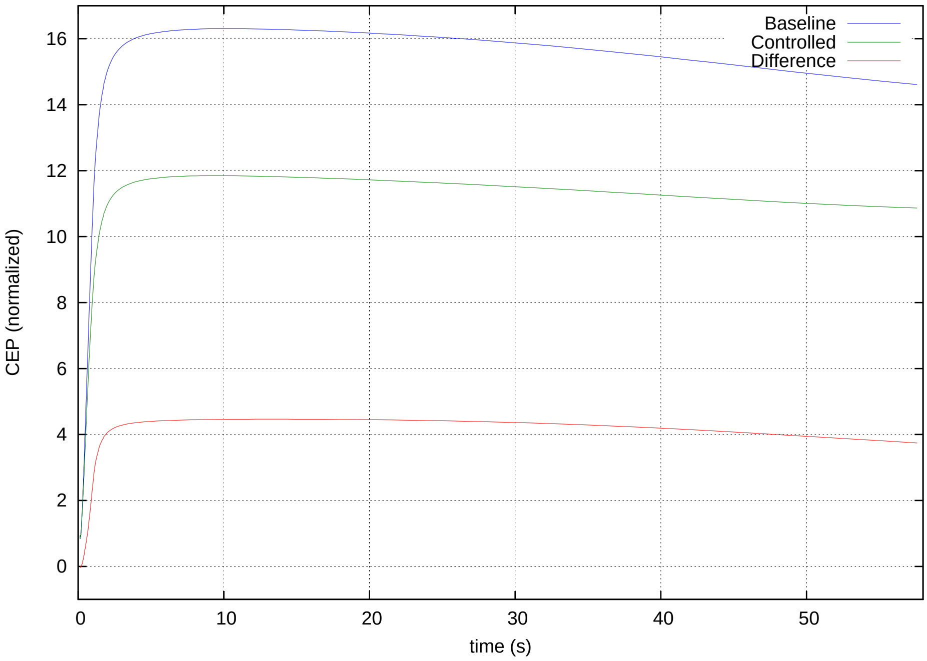

Figure 6.51: CEP for UKF/ESOQ Hybrid Estimator Controlled Rocket

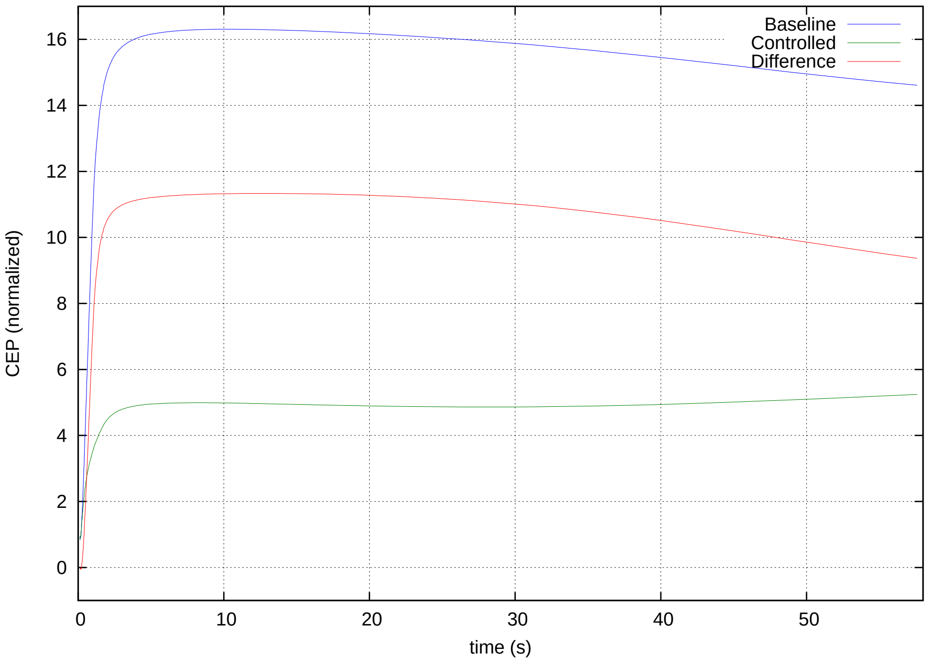

Figure 6.52: CEP for KF/ESOQ Hybrid Estimator Controlled Rocket

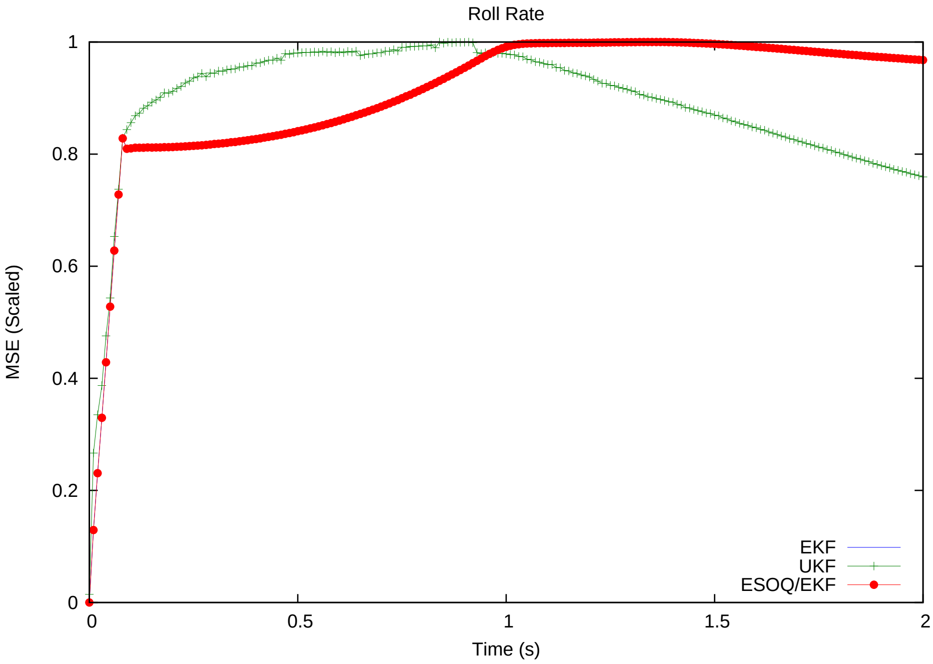

Figure 6.53: Roll Rate MSE for ESOQ Estimates as Input to an EKF

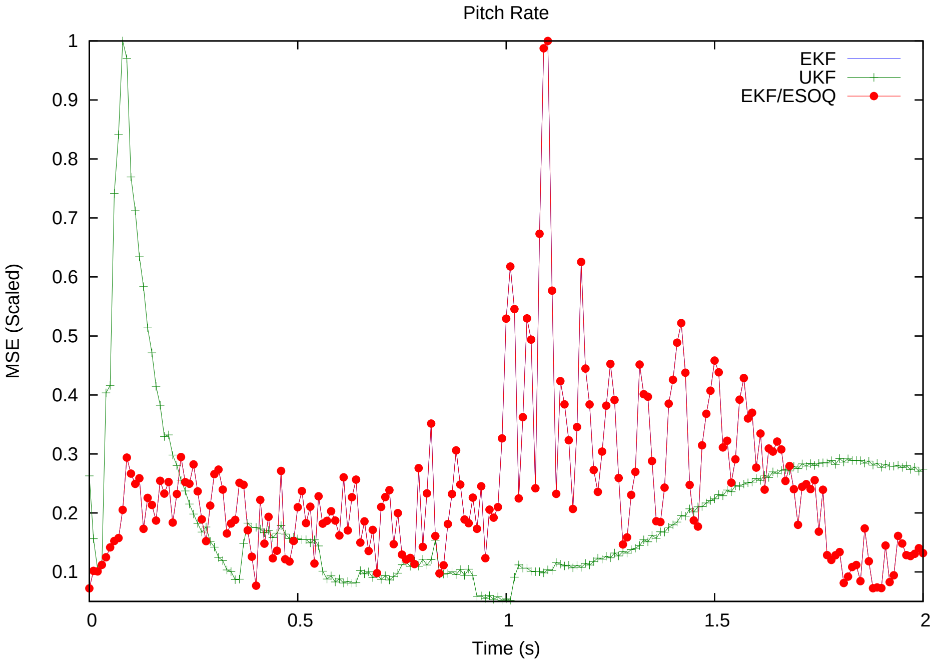

Figure 6.54: Pitch Rate MSE for ESOQ Estimates as Input to an EKF

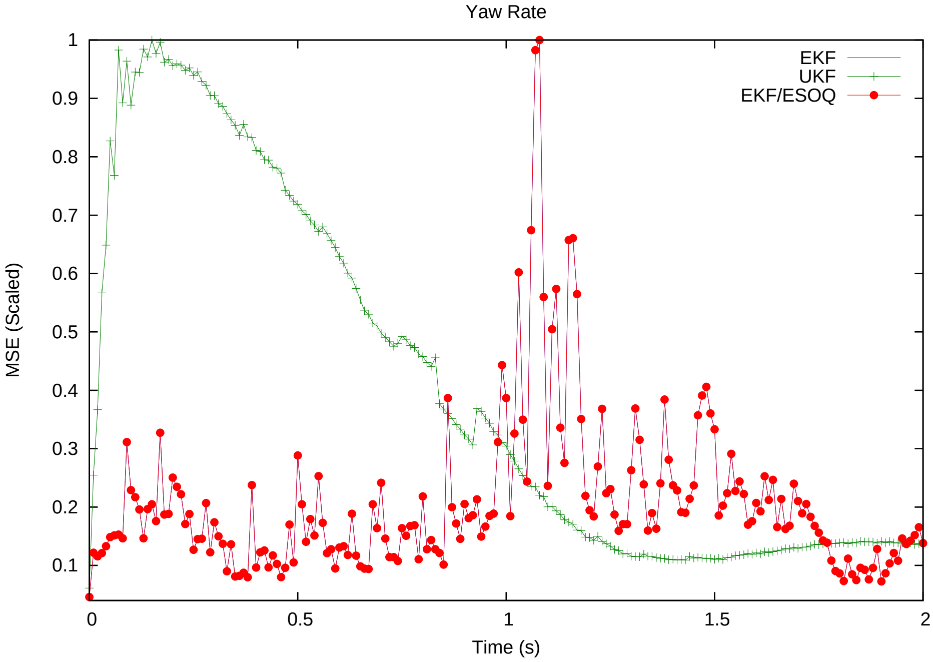

Figure 6.55: Yaw Rate MSE for ESOQ Estimates as Input to an EKF

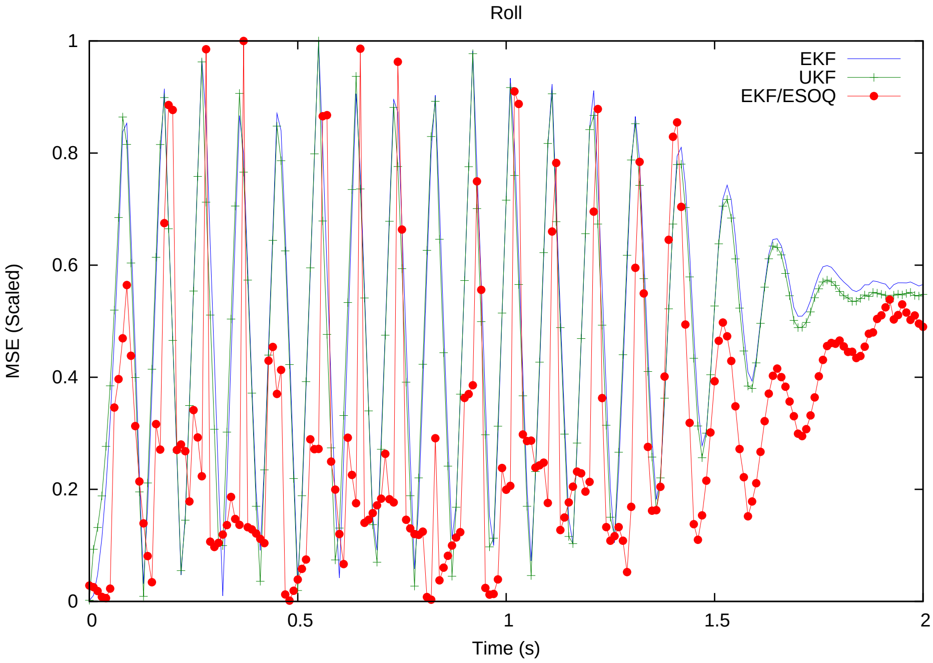

Figure 6.56: Roll Angle MSE for ESOQ Estimates as Input to an EKF

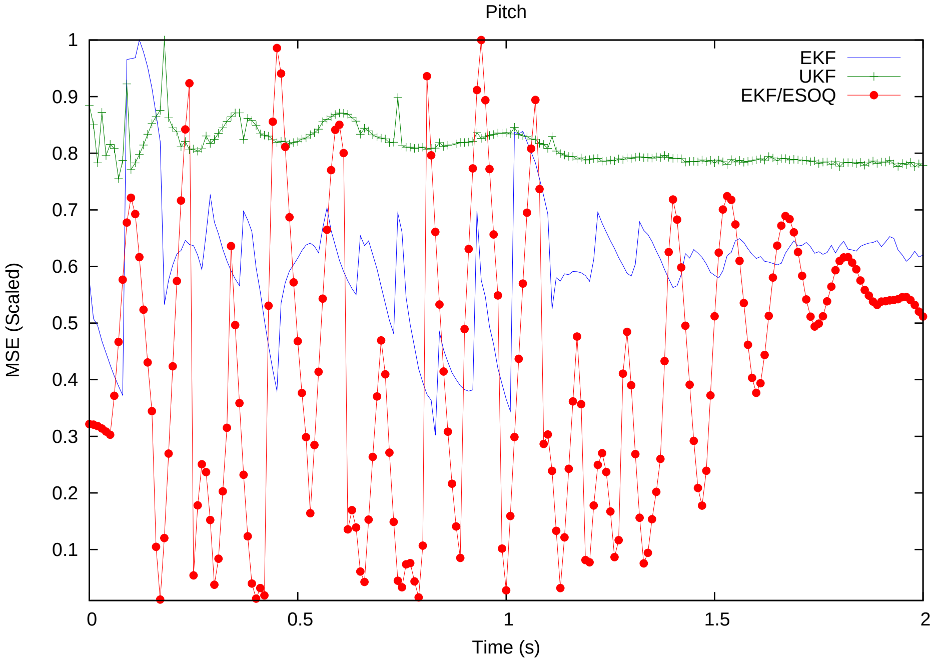

Figure 6.57: Pitch Angle MES for ESOQ Estimates as Input to an EKF

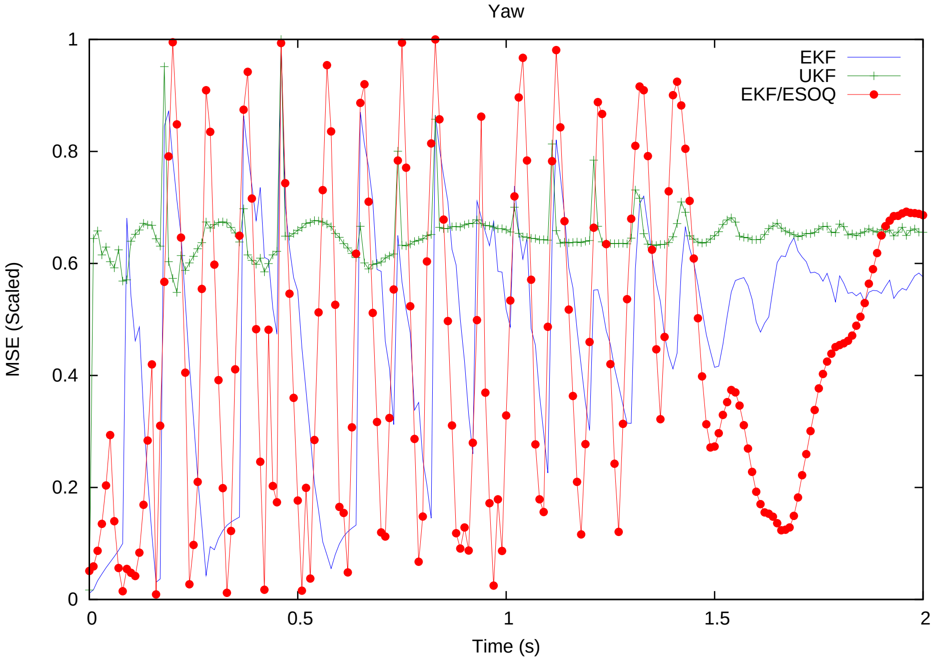

Figure 6.58: Yaw Angle MSE for ESOQ Estimates as Input to and EKF

In Figures 6.59-6.64 we show the MSEs for the UKF and EKF estimators as well as the EKF/ESOQ hybrid estimator, where the EKF is used to estimate angular rates, and the ESOQ algorithm is used to estimate angular position. As shown in Figures 6.59, 6.60, and 6.61, the roll, pitch, and yaw rate MSEs behave identically to those found in Figures 6.53-6.55. This is to be expected, since the EKF and UKF estimators are identical in both cases and the EKF/ESOQ hybrid is yielding purely EKF estimates for rate estimates. From this point forward UKF and pure EKF estimator MSEs will not be discussed, since they will be identical to those of Figures 6.53-6.58 for all remaining plots in which they appear. They appear in the plots to more easily compare their performances to the performance of the ESOQ estimators. In Figure 6.62 we show the roll angle MSEs. The EKF/ESOQ hybrid estimator has a fairly steady MSE at less than 0.1 units with an occasional spike for slightly more than 1 s. After 1 s the MSE begins to increase and settles to around 0.2 units while the EKF and UKF MSEs are greater than 0.55 units. The EKF/ESOQ hybrid MSE is worse than both the EKF and UKF MSEs for pitch angle as shown in Figure 6.63, holding steady at 1 unit. The yaw angle MSE is shown in Figure 6.64. The EKF/ESOQ hybrid oscillates between 0 and 1 units until the rocket's fins open at 1 s. The magnitude of the oscillations then begins to decrease until the MSE settles to 0.6 units.

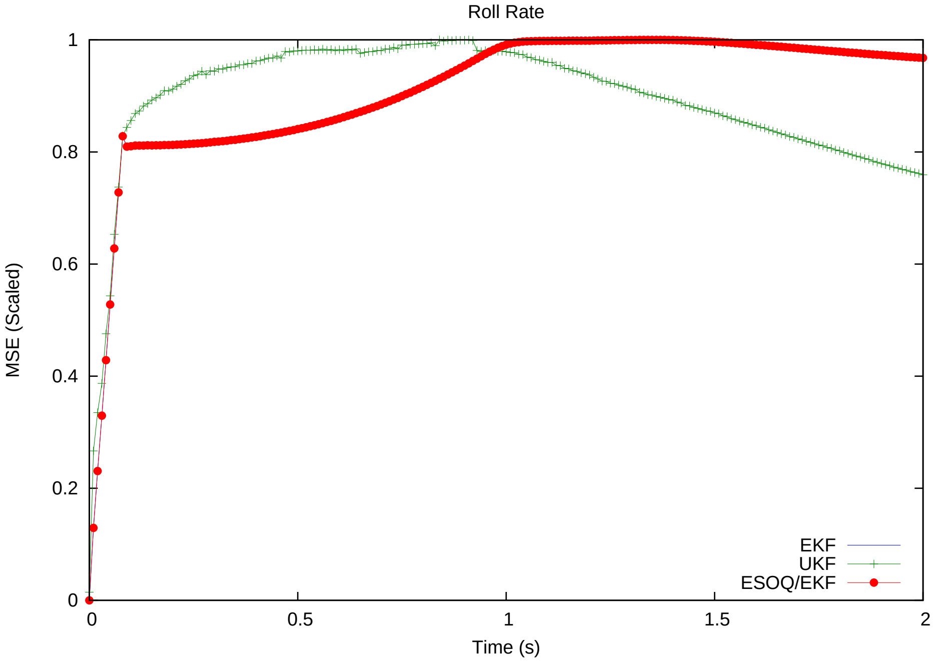

Figure 6.59: Roll Rate MSE for EKF/ESOQ Hybrid

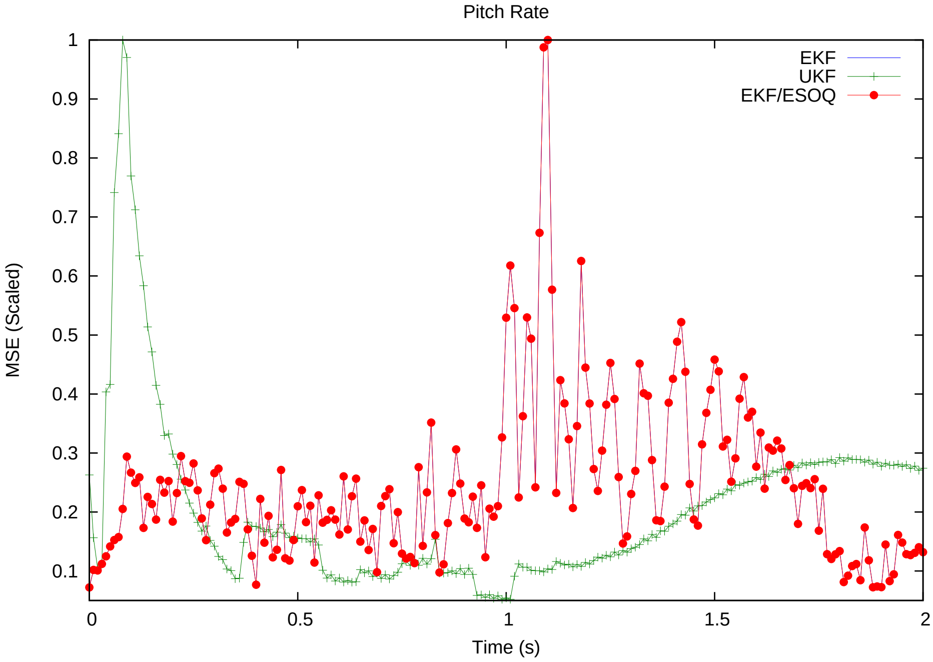

Figure 6.60: Pitch Rate MSE for EKF/ESOQ Hybrid

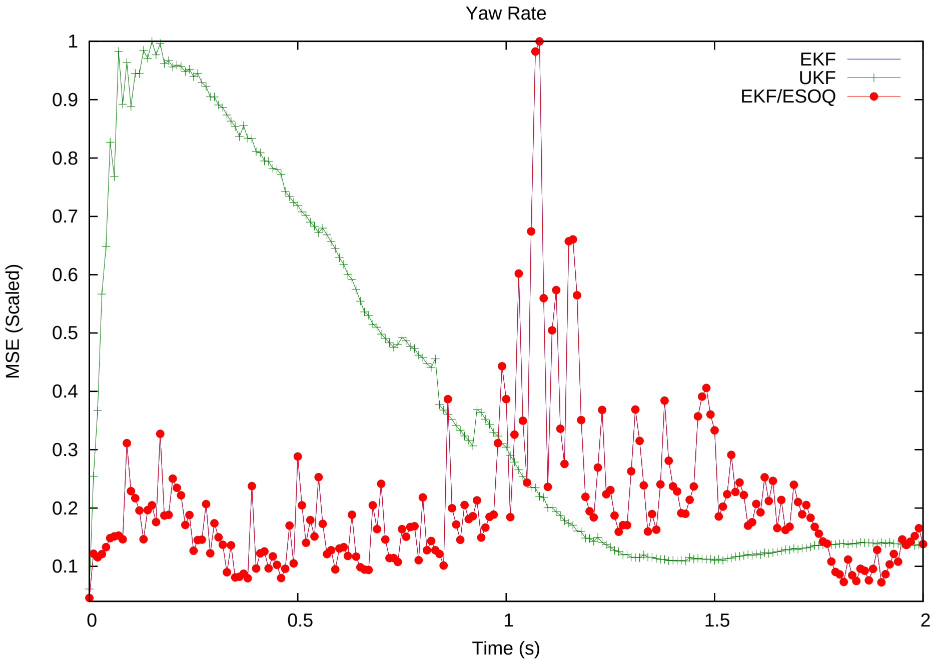

Figure 6.61: Yaw Rate MSE for EKF/ESOQ Hybrid

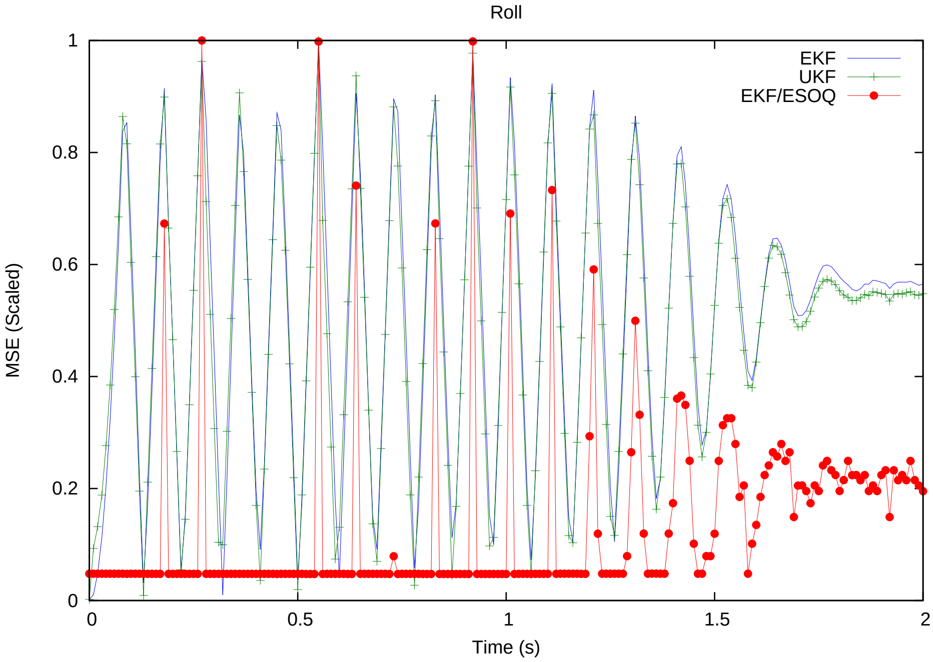

Figure 6.62: Roll Angle MSE for EKF/ESOQ Hybrid

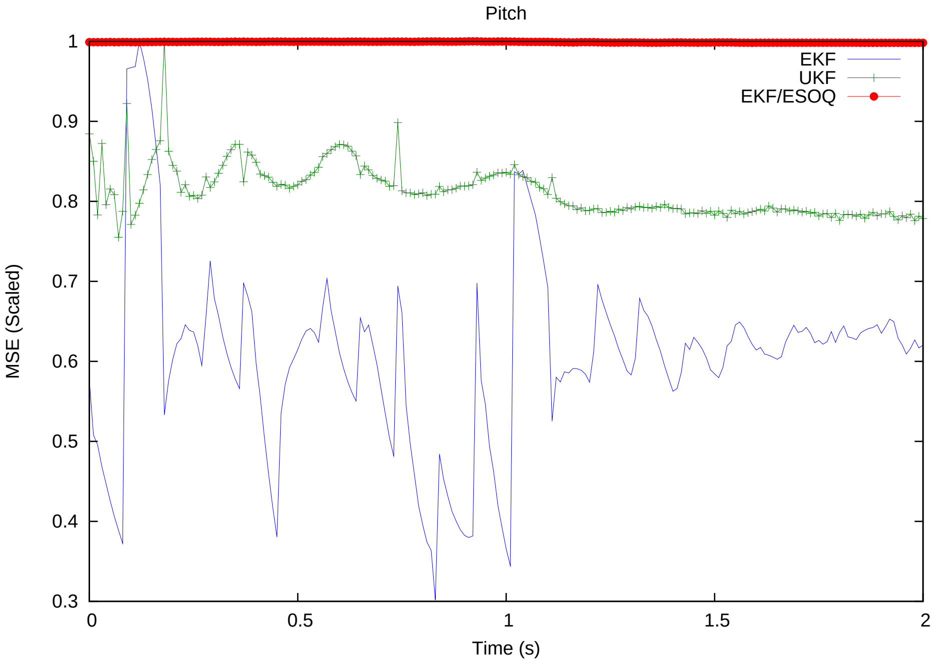

Figure 6.63: Pitch Angle MSE for EKF/ESOQ Hybrid

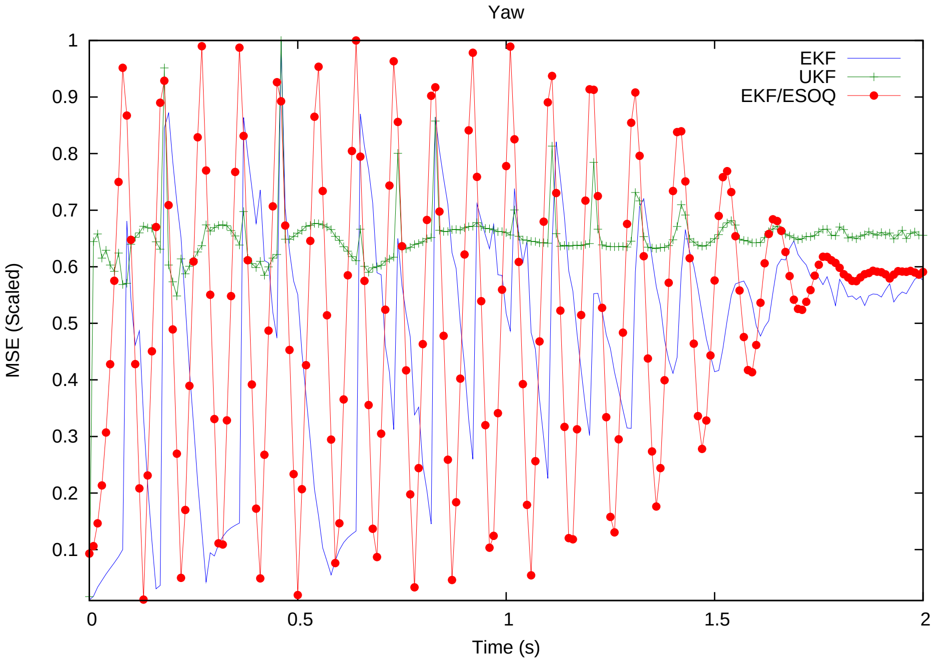

Figure 6.64: Yaw Angle MSE for EKF/ESOQ Hybrid

In Figures 6.65-6.70 we show the MSEs for the EKF, UKF, and UKF/ESOQ hybrid estimators. The UKF/ESOQ hybrid estimator uses the UKF to provide angular rate estimates and the ESOQ algorithm to provide angular position estimates. The MSE results of the UKF/ESOQ hybrid estimator are identical to those of the EKF/ESOQ hybrid.

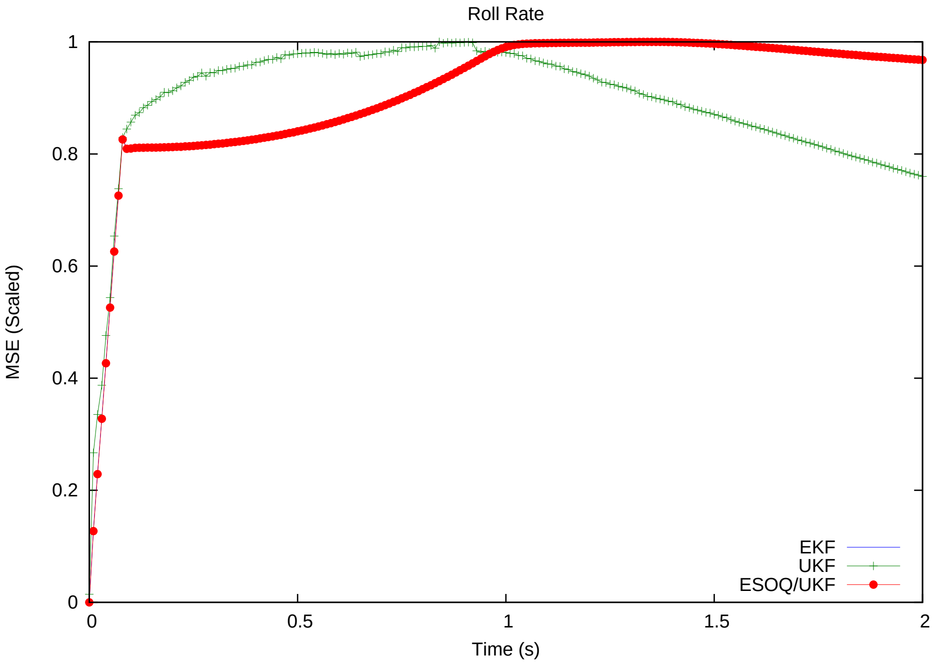

Figure 6.65: Roll Rate MSE for UKF/ESOQ Hybrid

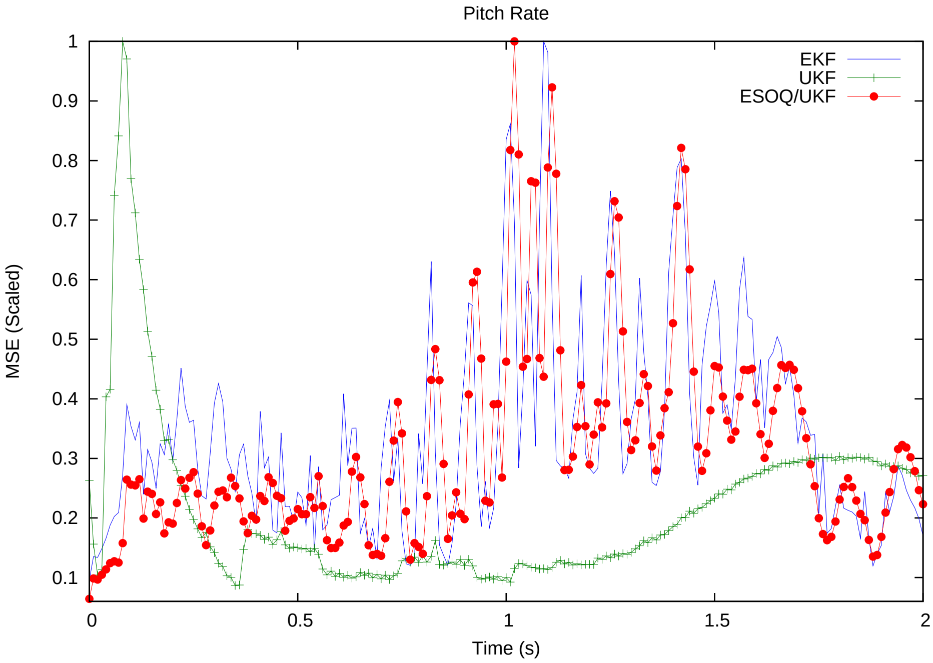

Figure 6.66: Pitch Rate MSE for UKF/ESOQ Hybrid

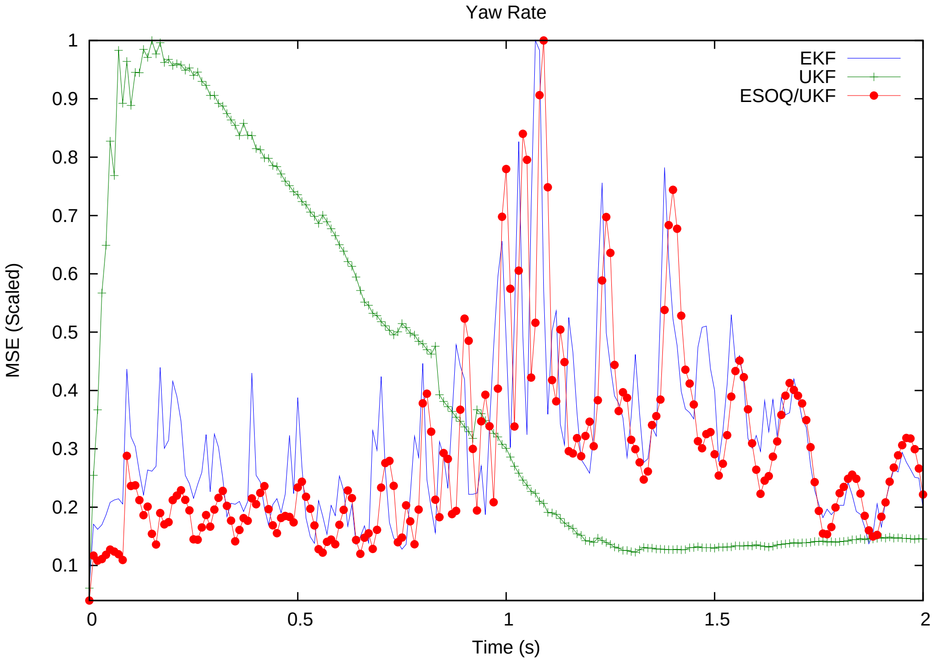

Figure 6.67: Yaw Rate MSE for UKF/ESOQ Hybrid

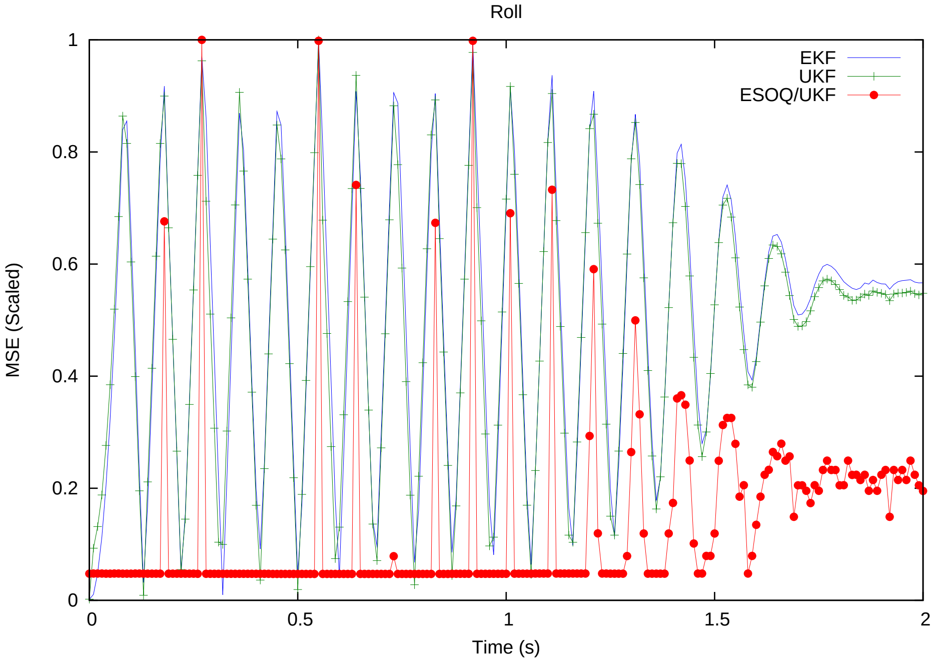

Figure 6.68: Roll Angle MSE for UKF/ESOQ Hybrid

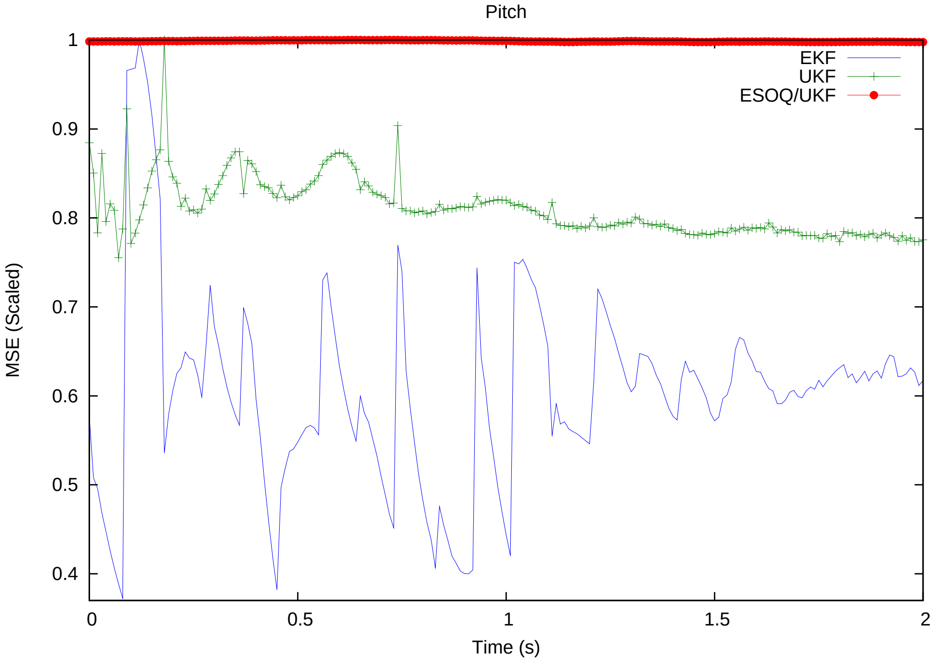

Figure 6.69: Pitch Angle MSE for UKF/ESOQ Hybrid

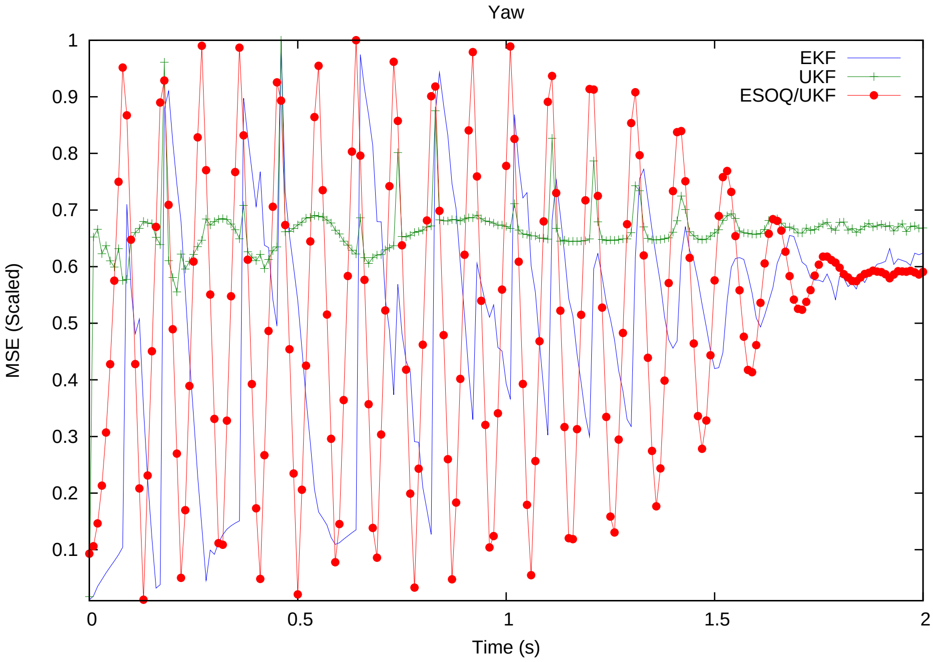

Figure 6.70: Yaw Angle MSE for UKF/ESOQ Hybrid

In Figures 6.71-6.76 we show the MSEs for the EKF, UKF, and KF/ESOQ hybrid estimators. The KF/ESOQ hybrid estimator uses the KF to provide angular rate estimates and the ESOQ algorithm to provide angular position estimates. We present roll rate MSEs in Figure 6.71. Although the roll rate is modeled as a known function, we show the roll rate MSE to validate our model. The MSE spikes when the rocket launches but falls shortly thereafter to remain just above 0 until the fins open. Once the fins open, the MSE slightly increases but is still under 0.1 units. As shown in Figure 6.72, the EKF/ESOQ hybrid has a worse MSE than both the EKF and UKF for pitch rate. While the MSE is initially near 0, it jumps to more than 0.4 units and remains near 0.5 units until the fins open at 1 s. At this point the MSE spikes to 1 unit and oscillates about 0.6 until finally falling to about 0.3 units. Yaw rate MSE for the KF/ESOQ hybrid is almost identical to the pitch rate MSE as shown in Figure 6.73. As we see in Figures 6.74, 6.75, and 6.76, the roll angle, pitch angle, and yaw angle MSEs, respectively, are identical to those of Figures 6.68-6.70 since the same estimator is used.

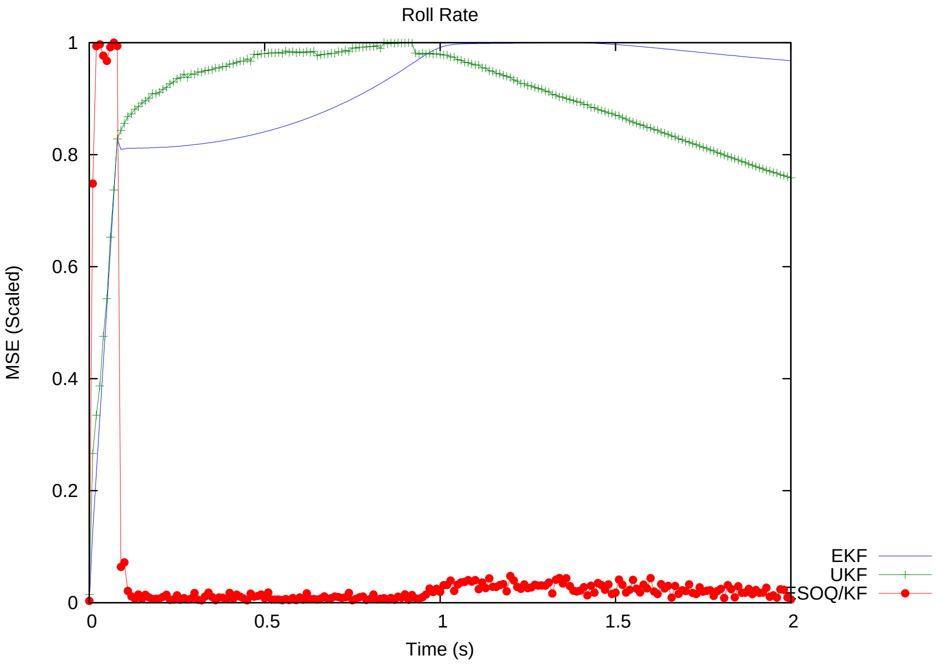

Figure 6.71: Roll Rate MSE for KF/ESOQ Hybrid

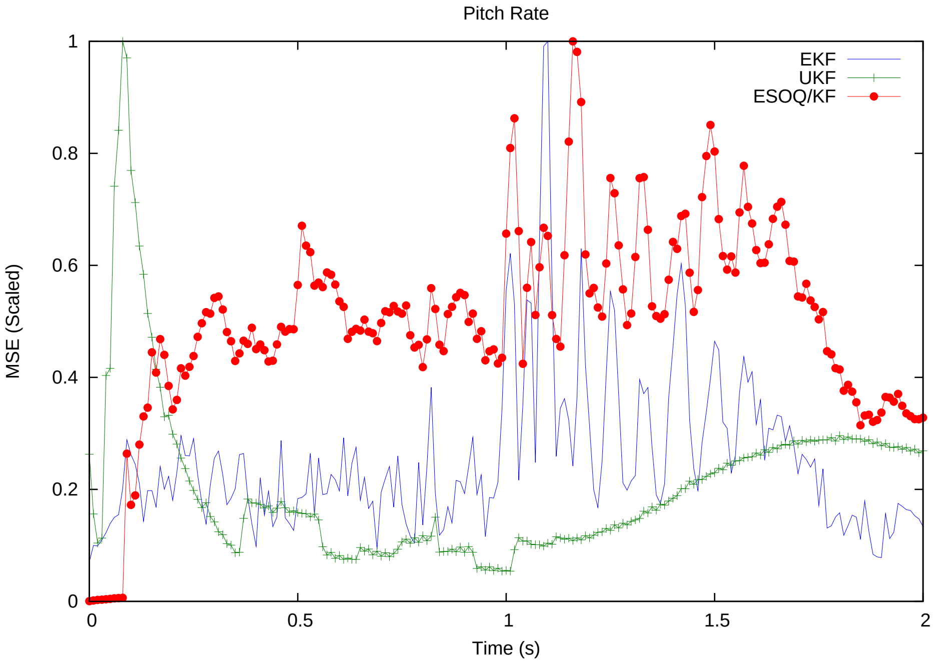

Figure 6.72: Pitch Rate MSE for KF/ESOQ Hybrid

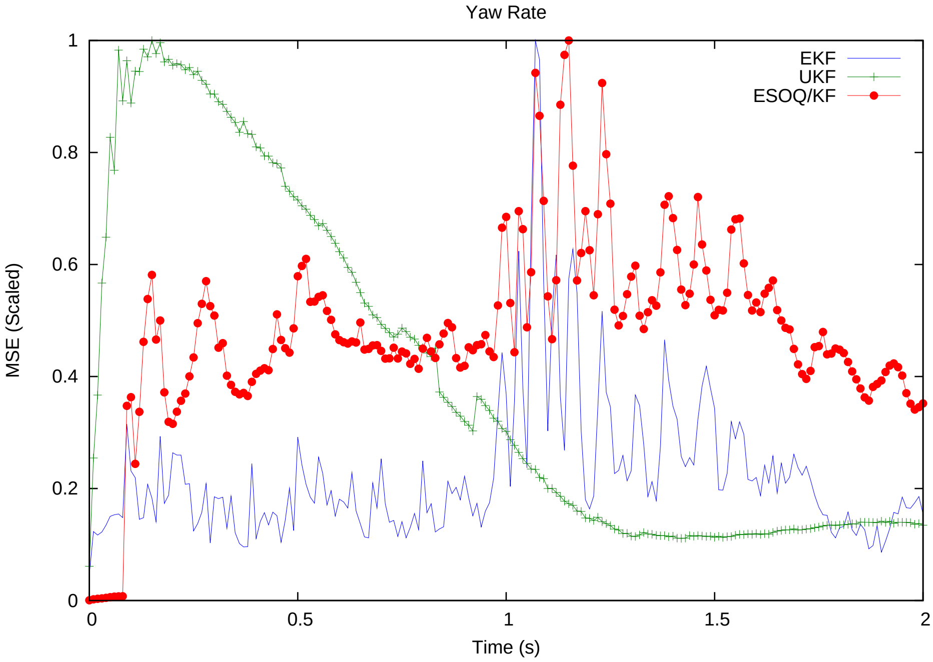

Figure 6.73: Yaw Rate MSE for KF/ESOQ Hybrid

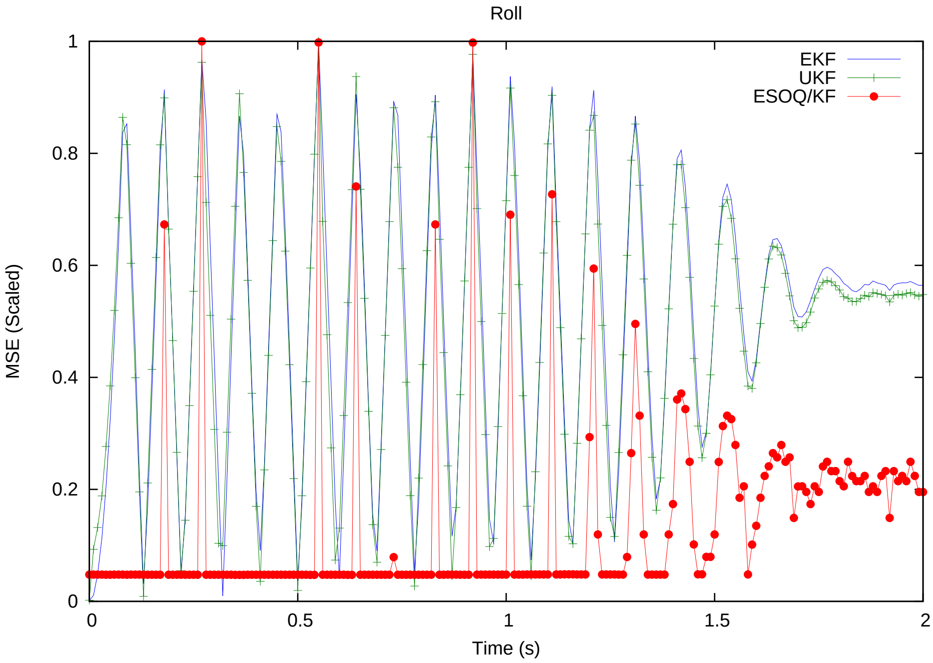

Figure 6.74: Roll Angle MSE for KF/ESOQ Hybrid

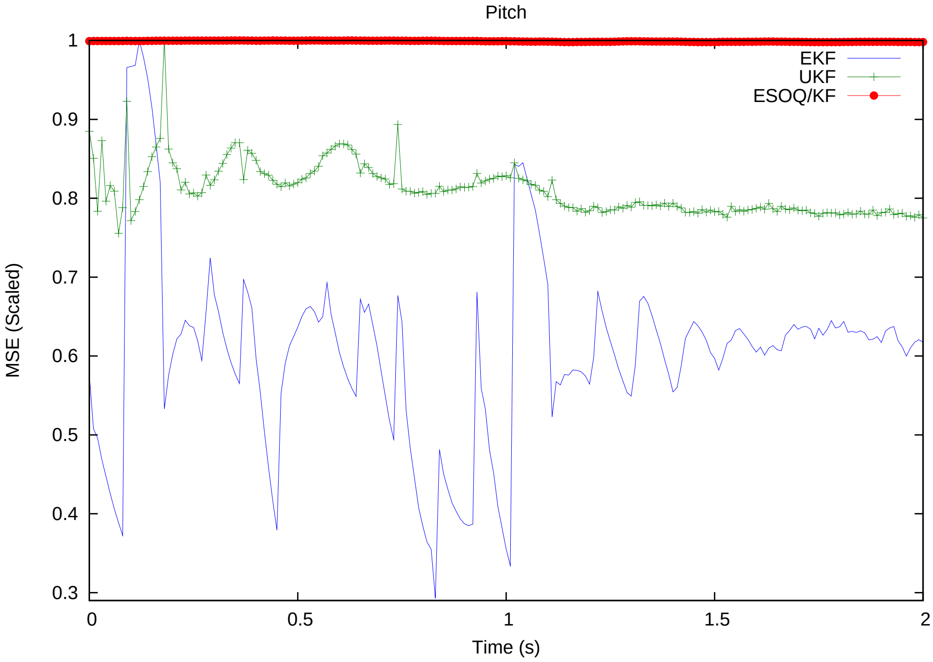

Figure 6.75: Pitch Angle MSE for KF/ESOQ Hybrid

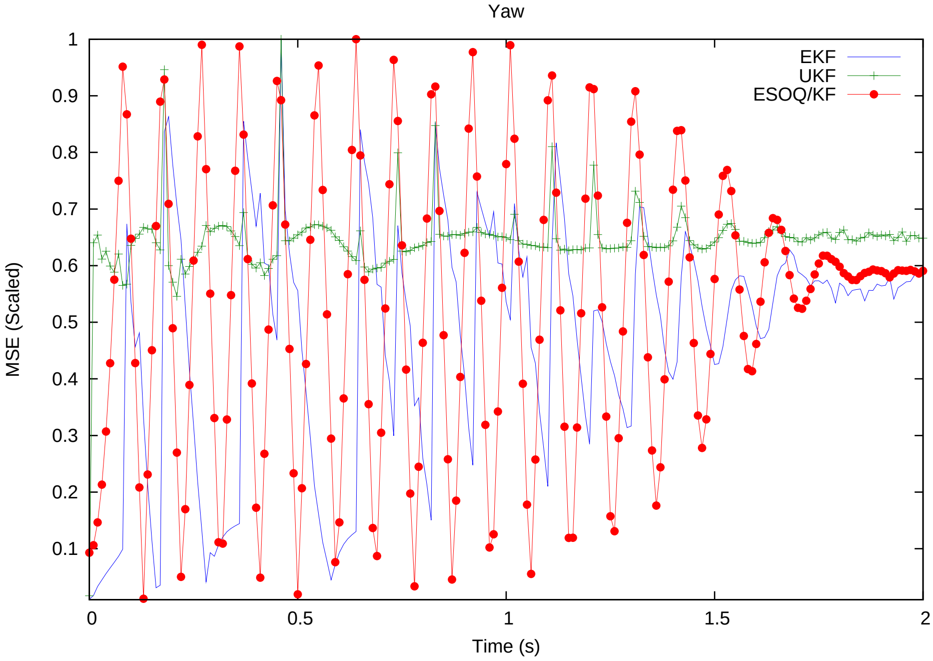

Figure 6.76: Yaw Angle MSE for KF/ESOQ Hybrid

6.6 Conclusions

We have presented the results of simulating a rocket that is controlled with estimates from a variety of estimators. The best performing estimators are the KF-based estimators, followed by the EKF-based estimators. The UKF-based estimators perform poorly, and the pure ESOQ estimators perform the worst of all. These results directly related to the ability of each estimator to accurately estimate roll rate. The control law uses the estimate of roll rate to determine if a lateral thruster will be in the correct position to improve the rocket's attitude when fired. As the roll rate estimate becomes more accurate, the control law is better able to yield the desired attitude. The KF-based estimators do not estimate roll rate; rather, they use a model of roll rate. Thus, the KF estimators yield the best results. The EKF- and UKF-based estimators estimate roll rate based on dynamic equations rather than using a model. The estimated roll rates are not as accurate as the modeled roll rates so the EKF and UKF do not perform as well as the KF. The worst-performing estimator, the ESOQ algorithm, does not provide a roll rate estimate at all. The ESOQ algorithm estimates roll angle, and Euler differentiation is used to estimate roll rate. The Euler differentiation poorly approximates roll rate so the ESOQ algorithm has the worst performance.

The importance of roll rate estimation explains why the KF with only rate gyros has better performance than other estimators with more sensors. The addition of angle gyros and magnetometers does nothing to improve estimates of roll rate. Angle gyros directly measure angular position, and magnetometers indirectly measure angular position. While the additional sensors do improve the performance of the EKF and UKF estimators, none of them approach the performance of the KF estimators.

Aside from the poor roll rate estimation, another factor affects the performance of the UKF estimator. A quaternion formulation of the system is used. The UKF algorithm is performed on vector space, which is closed under addition; however, rotation quaternions are not a member of vector space and are not closed under addition. Thus, the method the UKF uses to create sigma points does not guarantee that a valid attitude quaternion will result from the operation. To ensure that the quaternion component of the state vector is a valid rotation quaternion, it is normalized as necessary in the algorithm. This repeated normalization causes the mean and covariance of the random variable not to be preserved. A method to avoid this linearization is proposed by Cheon and Kim in [68] but was not used in this work.

The best improvement in controlled CEP over the baseline CEP is yielded by the Kalman filter with rate gyros and the Kalman filter in conjunction with the ESOQ algorithm. The Kalman filter with rate gyros, however, is a better choice than the KF/ESOQ hybrid, since the hybrid filter requires rate gyros, angle gyros, and a magnetometer. The KF/ESOQ hybrid is also more computationally expensive than the KF. The EKF performs essentially the same for each sensor package, reducing the baseline CEP almost as much as the KF. However, the EKF is computationally more expensive than the KF. The UKF performs poorly for each case as compared to the performance of the KF. The UKF's best performances are for the rate-gyro-only case and the rate-gyro and magnetometer case. The UKF's reduction in CEP is actually better for the rate gyro and magnetometer package than for the rate gyro and ideal vector package. This is because of the normalization of the quaternion component during estimation. The noise present on the magnetometer helps counteract the multiple normalizations. The UKF provides a worse CEP than the baseline CEP for the rate gyro, angle gyro, and magnetometer sensor suite. Both the angle gyros and magnetometer output models are nonlinear functions of the attitude quaternion, which exacerbates the effects of normalizing the quaternion. Thus, poor estimates result. The best-performing UKF based estimator is the UKF/ESOQ hybrid. Since the ESOQ algorithm is used to estimate attitude instead of the UKF, no quaternion normalization is needed. However, the results are still poor compared to the KF results.

Chapter 7
Conclusions and Future Work

In this chapter we summarize the work presented. In §7.1 we present the contributions this work has made to the field of state estimation. We conclude the chapter with §7.2 in which we pose questions that directly follow from the work presented.

7.1 Contributions

In this work, we have made several contributions. We began by presenting various algorithms that find solutions to Wahba's problem. We evaluated these algorithms, presenting their advantages and disadvantages, to choose the appropriate algorithm for the ballistic rocket problem. Next, we presented several algorithms that were designed to estimate the constant biases of a magnetometer on a satellite. After reviewing each algorithm, we selected the algorithm best suited for estimating the constant biases of a magnetometer on a rocket in a launch mechanism. These algorithms had never previously been applied to a ballistic projectile since magnetometers are usually found on orbiting satellites. In chapter 4, we presented several algorithms for controlling a rocket using reaction jets. After evaluating their merits, we selected the algorithm most appropriate for a rocket controlled solely by reaction jets for the first few seconds of flight.

This work has also made several contributions to the field of state estimation of a ballistic rocket. We have addressed the issue of state estimation of a rocket that is controlled solely with a ring of lateral jets. While the control of such a rocket has been previously addressed, state estimation has not. We also focused on state estimation during the first few seconds of flight in a harsh environment. During this time the rocket is subject to many error sources including blow-back and tip-off errors. While previous work has addressed the issue of state estimation of projectiles during later stages of flight, the first few seconds of flight had been ignored. We also compared the effect that various MEMS sensor suites had on the state estimation problem. We modeled an angle gyro and implemented it in our simulations. Angle gyros, new developmental sensors, have never previously been simulated in such a manner. A magnetometer also augmented some sensor suites. The addition of a magnetometer to a rocket equipped only with inertial sensors had never previously been done.

We used various state estimators in new ways in this work. We compared the performance of the KF, EKF, UKF, and ESOQ algorithms for estimating the states of a rotating rocket and evaluated which estimator performed best. The ESOQ algorithm, which provides a solution to Wahba's problem, had never previously been applied to a vehicle other than an orbiting satellite. The use of the ESOQ algorithm was made possible by adding a magnetometer and angle gyros to the rocket. We also used a solution to Wahba's problem (the ESOQ algorithm) in conjunction with Kalman filter based estimators, the EKF and UKF. No previous work has combined any solution of Wahba's problem with Kalman filtering for estimating the states of a rocket.

7.2 Future Work

Some questions concerning this work have yet to be answered. One area worth exploring is the addition of more reaction jets to the rocket. What would be the affect on accuracy if more reaction jets were available to apply torques? It is also of interest to study the effects of multiple reaction jets that can be fired multiple times. If the rocket was equipped with more memory capability, more possibilities would be available. A magnetic field model of the earth could be stored on the rocket, which would impact the accuracy of a magnetometer on a rocket. Also, better and more complex sensor calibration algorithms could be implemented. More memory would also allow a more accurate model of the rocket to be stored onboard. This leads to the question of how would a more accurate model affect the performance of the various state estimators presented in this work.

Bibliography

[1] G. Wahba, "A least squares estimate of spacecraft attitude," *SIAM Review*, vol. 7, no. 3, p. 409, Jul 1965.

[2] J. L. Farrell, J. C. Stuelpnagel, R. H. Wessner, J. Velman, and J. E. Brock, "Problem 65-1, a least squares estimate of satellite attitude," *SIAM Review*, vol. 8, no. 3, pp. 384–386, Jul 1966.

[3] P. B. Davenport, "A vector approach to the algebra of rotations with applications," NASA, Goddard Space Flight Center Greenbelt, MD, Tech. Rep. TN D-4696, Aug 1968.

[4] J. R. Wertz, Ed., *Spacecraft Attitude Determination and Control.* D. Reidel Publishing Company, 1978.

[5] I. Y. Bar-Itzhack, "REQUEST: A recursive QUEST algorithm for sequential attitude determination," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 5, pp. 1034–1038, 1996.

[6] M. D. Shuster, "Maximum likelihood estimation of spacecraft attitude," *Journal of Astronautical Sciences*, vol. 37, no. 1, pp. 79–88, Jan-Mar 1989.

[7] ——, "A simple Kalman filter and smoother for spacecraft attitude," *Journal of the Astronautical Sciences*, vol. 37, no. 1, pp. 89–106, 1989.

[8] M. L. Psiaki, "Attitude determination filtering via extended quaternion estimation," *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 2, pp. 206–214, 2000.

[9] D. Mortari, "Energy approach algorithm for attitude determination from vector observations," *Journal of the Astronautical Sciences*, vol. 45, no. 1, pp. 41–55, 1997.

[10] F. L. Markley, "Attitude determination using vector observations and the singular value decomposition," *The Jounal of the Astronautical Sciences*, vol. 36, no. 3, pp. 245–258, Jul-Sep 1988.

[11] ——, "Attitude determination using vector observations: A fast optimal matrix algorithm," *The Journal of Astronautical Sciences*, vol. 41, no. 2, pp. 261–280, Apr-Jun 1993.

[12] ——, "New quaternion attitude estimation method," *Journal of Guidance, Control, and Dynamics*, vol. 17, no. 2, pp. 407–409, Mar-Apr 1994.

[13] D. Mortari, "Euler-q algorithm for attitude determination from vector observations," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 3, pp. 328–334, 1998.

[14] ——, "ESOQ: A closed-form solution to the Wahba problem," *Journal of the Astronautical Sciences*, vol. 45, no. 2, pp. 195–204, Apr-Jun 1997.

[15] ——, “ESOQ2 single-point algorithm for fast optimal spacecraft attitude determination,” *Advance in the Astronautical Sciences*, vol. 95, no. II, pp. 817–826, 1997.

[16] F. L. Markley and I. Y. Bar-Itzhack, “Unconstrained optimal transformation matrix,” *IEEE Transactions of Aerospace and Electronic Systems*, vol. 34, no. 1, pp. 338–340, Jan 1998.

[17] C. M. Roithmayr, “Contributions of spherical harmonics to magnetic and gravitational fields,” NASA, Tech. Rep. TM-2004-213007, Mar 2004, accessed Dec. 12, 2006. [Online]. Available: http://hdl.handle.net/2002/13272

[18] S. McLean, “IAGA division V-Mod geomagnetic field modeling,” International Association of Geomagnetism & Aeronomy and the International Union of Geodesy & Geophysics, accessed Dec 18, 2006. [Online]. Available: http://www.ngdc.noaa.gov/IAGA/vmod/

[19] S. McLean, S. Maus, D. Dater, S. Macmillan, V. Lesur, and A. Thomson, “The US/UK world magnetic model for 2005-2010,” NOAA, Tech. Rep. NESDIS/NGDC-1, 2004, accessed Dec 18, 2006. [Online]. Available: http://www.ngdc.noaa.gov/seg/WMM/DoDWMM.shtml

[20] J. A. McClendon, Jr., “Full three axis calibration technique,” Master’s thesis, Auburn University, 2002.

[21] R. Alonso and M. D. Shuster, “Attitude-independent magnetometer-bias determination: A survey,” *Journal of the Astronautical Sciences*, vol. 50, no. 4, pp. 453–475, Oct-Dec 2002.

[22] ——, “Complete linear attitude independent magnetometer calibration,” *The Journal of the Astronautical Sciences*, vol. 50, no. 4, pp. 477–490, Oct-Dec 2002.

[23] M. S. Hodgart and P. Tortora, “A recursive optimised algorithm for in-flight magnetometer calibration without knowledge of attitude,” in *Proceedings of the* 4^{th} *ESA International Conference on Spacecraft Guidance, Navigation and Control Systems*, Oct 18-21 1999, pp. 167–172.

[24] P. Tortora, Y. Oshman, and F. Santoni, “Attitude independent estimation of spacecraft angular rate using geomagnetic field observations,” in *Proceedings of the 2003 IEEE Aerospace Conference*, vol. 6, Mar 8-15 2003, pp. 6_2636–6_2645.

[25] J.-I. Lee and I.-J. Ha, “Autopilot design for highly maneuvering STT missiles via singular perturbation-like technique,” *IEEE Transactions on Control Systems Technology*, vol. 7, no. 5, pp. 527–541, Sep 1999.

[26] J. Y. Choi and D. K. Chwa, “Adaptive control based on parametric affine model for tail-controlled missiles,” in *Proceedings of the* 39^{th} *IEEE Conference on Decision and Control*, vol. 2. IEEE, Dec 12-15 2000, pp. 1471–1476.

[27] F.-K. Yeh, K.-Y. Cheng, and L.-C. Fu, “Variable structure-based nonlinear missile guidance/autopilot design with highly maneuverable actuators,” *IEEE Transactions on Control Systems Technology*, vol. 12, no. 6, pp. 944–949, Nov 2004.

[28] S. E. Lyshevski, “Space transformation method in control of agile interceptors and missiles with advanced microelectromechanical actuators,” in *Proceedings of the* 42^{nd} *IEEE Conference on Decision and Control*, vol. 5. IEEE, Dec 9-12 2003, pp. 5432–5437.

[29] A. Thukral and M. Innocenti, "A sliding mode missile pitch autopilot synthesis for high angle of attack maneuvering," *IEEE Transactions on control Systems Technology*, vol. 6, no. 3, pp. 359–371, May 1998.

[30] T. L. Song and S. J. Shin, "Time-optimal impact angle control for vertical plane engagements," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 2, pp. 738–742, Apr 1999.

[31] A. V. Savkin, P. N. Pathirana, and F. A. Faruqi, "The problem of precision missile guidance: LQR and H^∞ control frameworks," in *Proceedings of the* 40^{th} *IEEE Conference on Decision and Control*, vol. 2. IEEE, Dec 4-7 2001, pp. 1535–1540.

[32] C. Kwan, R. Xu, W. Liu, R. Tan, and L. Haynes, "Nonlinear control of missile dynamics," in *Proceedings of the* 37^{th} *IEEE Conference on Decision and Control*, vol. 4. IEEE, Dec 16-18 1998, pp. 4685–4690.

[33] L.-C. Fu, C.-W. Tsai, and F.-K. Yeh, "A nonlinear missile guidance controller with pulse type input devices," in *Proceedings of the 1999 American Control Conference*, vol. 6. IEEE, Jun 2-4 1999, pp. 3753–3757.

[34] Y. S. Choi, H. C. Lee, and J. W. Choi, "Autopilot design for agile missile with aerodynamic fin and side thruster," in *SICE 2003 Annual Conference*, vol. 2. IEEE, Aug 4-6 2003, pp. 1476–1481.

[35] H. C. Lee, J. W. Choi, T. L. Song, and C. H. Song, "Agile missile autopilot design via time-varying eigenvalue assignment," in *ICARCV 2004* 8^{th} *Control, Automation, Robotics and Vision Conference*, vol. 3, Dec 6-9 2004, pp. 1832–1837.

[36] M. B. McFarland and A. J. Calise, "Adaptive nonlinear control of agile antiair missiles using neural networks," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 5, pp. 749–756, Sep 2000.

[37] M. Innocenti and A. Thukral, "Robustness of a variable structure control system for maneuverable flight vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 12, no. 2, pp. 377–383, Mar-Apr 1997.

[38] A. Thukral and M. Innocenti, "Variable structure autopilot for high angle of attack maneuvers using on-off thrusters," in *Proceedings of the* 33^{rd} *IEEE Conference on Decision and Control*, vol. 4. IEEE, Dec 14-16 1994, pp. 3850–3851.

[39] W. Tan, A. K. Packard, and G. J. Balas, "Quasi-LPV modeling and LPV control of a generic missile," in *Proceedings of the American Control Conference*, vol. 5, Jun 28-30 2000, pp. 3692–3696.

[40] H. S. Nam, S.-H. Kima, C. Song, and J. Lyou, "A robust nonlinear control approach to missile autopilot design," in *Proceedings of the* 40^{th} *SICE Annual Conference. International Session Papers*. IEEE, Jul 25-27 2001, pp. 310–314.

[41] J.-Y. Yu, Y.-A. Zhang, and W.-J. Gu, "An approach to integrated guidance/autopilot design for missiles based on terminal sliding mode control," in *Proceedings of 2004 International Confernce on Machine Learning and Cybernetics*, vol. 1, Aug 26-29 2004, pp. 610–615.

[42] H. Zhao, W. Gu, Y. Hu, and C. Pan, "Second-order sliding mode control for aerodynamic missiles using backstepping design," in *Proceedings of the* 5th *World Congress on Intelligent Control and Automation*, vol. 6, Jun 15-19 2004, pp. 5471–5474.

[43] R. Hindman and W. M. Shell, "Missile autopilot design using adaptive nonlinear dynamic inversion," in *Proceedings of the 2005 American Control Conference*, vol. 6, Jun 8-10 2005, pp. 3918–3919.

[44] D. Malloy and B. C. Chang, "Regulator design for linear parameter varying systems using dynamic inversion," in *Proceedings of the* 35th *Conference on Decision and Control*, vol. 2. IEEE, Dec 11-13 1996, pp. 2271–2276.

[45] Y.-Y. Chen, Bor-SenChen, and C.-S. Tseng, "Adaptive fuzzy mixed H_2/H_∞ lateral control of nonlinear missile systems," in *Proceedings of the* 126th *IEEE International Conference on Fuzzy Systems*, vol. 1. IEEE, May 25-28 2003, pp. 512–516.

[46] D. Dumlu and K. Özcaldiran, "Design of a fixed H_∞ controller for a missile," in *Proceedings of the 1998 IEEE International Conference on Control Applications*, vol. 2. IEEE, Sep 1-4 1998, pp. 985–989.

[47] I. Astrov, A. Pedai, and E. Rüstern, "Simulation of two-rate adaptive hybrid control with neural and neuro-fuzzy networks for stochastic model of missile autopilot," in *Proceedings of the* 5th *World Congress on Intelligent Control and Automation.* IEEE, Jun 15-19 2004, pp. 2603–2607.

[48] A. Elsakann, A. Bahnasawi, and M. Elamir, "High angle of attack maneuvering via intelligent approaches," in *Proceedings of the* 46th *IEEE International Midwest Symposium on Circuits and Systems*, vol. 3. IEEE, Dec 27-30 2003, pp. 1039–1042.

[49] Y. Yuan, Y. Feng, and W. Gu, "Fuzzy model reference learning control for aircraft pitch autopilot design," in *ICARCV 2004* 8th *Control, Automation, Robotics, and Vision Conference*, vol. 3, Dec 6-9 2004, pp. 1923–1927.

[50] T. Kuhn, "Aspects of pure and satellite-aided inertial navigation for gun-launched munitions," in *IEEE PLANS, Position Location and Navigation Symposium*, 2004, pp. 327–336.

[51] J. Thienel and R. M. Sanner, "A coupled nonlinear spacecraft attitude controller and observer with an unknown constant gyro bias and gyro noise," *IEEE Transactions on Automatic Control*, vol. 48, no. 11, pp. 2011–2015, Nov 2003.

[52] D. Gebre-Egziabher, G. H. Elkaim, J. D. Powell, and B. W. Parkinson, "A gyro-free quaternion-based attitude determination system suitable for implementation using low cost sensors," in *Position Location and Navigation Symposium, IEEE 2000*, Mar 13-16 2000, pp. 185–192.

[53] R. D. Weil and K. A. Wise, "Blended aero & reaction jet missile autopilot design using VSS techniques," in *Proceedings of the* 30th *Conference on Decision and Control*, vol. 3. IEEE, Dec 11-13 1991, pp. 2828–2829.

[54] E. Jahangir, "Time-optimal attitude control of a spinning missile," Ph.D. dissertation, University of Michigan, 1990.

[55] E. Jahangir and R. M. Howe, “A two-pulse scheme for the time-optimal attitude control of a spinning missile,” in *AIAA Guidance, Navigation and Control Conference*, Aug 20-22 1990, pp. 550–560.

[56] ——, “A scheme to generate thruster firing times for the time-optimal reorientation of a spinning missile,” in *29^{th} Aerospace Sciences Meeting*, Jan 7-10 1991, pp. 1–11.

[57] ——, “Time-optimal attitude control scheme for a spinning missile,” *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 2, pp. 346–353, Mar-Apr 1993.

[58] T. Jitpraphai, B. Burchett, and M. Costello, “A comparsion of different guidance schemes for a direct fire rocket with a pulse jet control mechanism,” U.S. Army Research Laboratory, Tech. Rep. ARL-CR-493, Apr 2002.

[59] T. Jitpraphai and M. Costello, “Dispersion reduction of a direct fire rocket using lateral pulse jets,” *Journal of Spacecraft and Rockets*, vol. 38, no. 6, pp. 929–936, Nov-Dec 2001.

[60] B. Burchett and M. Costello, “Model predictive lateral pulse jet control of an atmospheric rocket,” *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 5, pp. 860–867, Sep-Oct 2002.

[61] A. Anderson, D. Bittle, R. Dean, G. Flowers, J. Hester, and A. Hodel, “Investigation of mal-launch correction in spin-stabilized rockets,” in *Proceedings of IEEE SoutheastCon*, Mar 2007, pp. 267–272.

[62] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. 2715 North Charles Street, Baltimore, MD 21218-4363: The John Hopkins University Press, 1996.

[63] A. M. Shkel, C. Akar, and C. Painter, “Two types of micromachined vibratory gyrscopes,” in *Proceedings of Fourth IEEE Conference on Sensors*, Oct-Nov 2005, pp. 531–536.

[64] C. Painter and A. Shkel, “Active structural error suppression in MEMS vibratory gyroscopes,” in *Proceedings of the First IEEE International Conference on Sensors*, vol. 1, no. 2, Jun 12-14 2002, pp. 1089–1094.

[65] C. C. Painter, “Micromachined vibratory gyroscopes with imperfections,” Ph.D. dissertation, University of California, Irvine, 2005.

[66] S. Julier and J. Uhlmann, “A new extension of the Kalman filter to nonlinear systems,” in *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, Orlando, FL*, 1997. [Online]. Available: citeseer.ist.psu.edu/julier97new.html

[67] A. E. Taub and M. A. Thomas, “Conficence intervals for CEP when the errors are elliptical normal,” Naval Surface Weapons Center, Tech. Rep. NSWC TR 86-205, Nov 1983.

[68] Y.-J. Cheon and J.-H. Kim, “Unscented filtering in a unit quaternion space for spacecraft attitude estimation,” in *IEEE International Symposium on Industrial Electronics*, Jun 4-7 2007, pp. 66–71.

[69] J. B. Kuipers, *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality.* Princeton, New Jersey: Princeton University Press, 1999.

[70] W. F. Phillips, C. E. Hailey, and G. A. Gebert, “Review of attitude representations used for aircraft kinematics,” *Journal of Aircraft*, pp. 718–737, Jul/Aug 2001.

[71] W. J. Thompson, *Atlas for Computing Mathematical Functions.* New York: John Wiley & Sons, INC., 1997.

[72] W. D. Parkinson, *Introduction to Geomagnetism.* Edinburgh, Scottland: Scottish Academic Press, Ltd., 1984.

[73] R. F. Stengel, *Optimal Control and Estimation.* New York: Dover Publications, Inc., 1994.

[74] S. Julier and J. K. Uhlmann, “A general method for approximating nonlinear transformations of probability distributions,” Department of Engineering Science, University of Oxford, Oxford, Tech. Rep. OX1 3PI UK, Nov 1996.

[75] E. A. Wan and R. van der Merwe, “The unscented Kalman filter for nonlinear estimation,” in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000*, Oct 1-4 2000, pp. 153–158.

[76] R. van der Merwe and E. A. Wan, “The square-root unscented Kalman filter for state and parameter-estimation,” in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, May 7-11 2001, pp. 3461–3464.

[77] D. Rick, “Deriving the haversine formula,” The Math Forum, April 1999, accessed Dec 18, 2006. [Online]. Available: http://mathforum.org/library/drmath/view/51897.html

[78] E. D. Kaplan, Ed., *Understanding GPS Principles and Applications.* Boston: Artech House, 1996.

Appendices

Appendix A

Rotation Sequences

A common problem that arises in modeling the dynamics of a moving body is changing from one reference frame to another. Many reference frames exist, with the two most common being the inertial reference frame and the body-fixed reference frame. An inertial reference frame has its axes fixed relative to the motion of the body. Its positive x-axis points north, its positive y-axis points east, and its positive z-axis points down. A body-fixed reference frame usually has its origin located at the body's center of gravity. Its positive x-axis points toward the front of the vehicle, such as toward the nose of an airplane. The positive y-axis points to the vehicle's right, and the positive z-axis points down. A representation of these two coordinate frames is shown in Figure 2.1. Converting between the inertial and body frames or any other reference frame involves sequences of rotations. The most common way to represent these rotation sequences is either via Euler angles or quaternions. References [69] and [70] give a detailed treatment of the mathematics and applications of quaternions as well as Euler angles.

A.1 Euler Angles

The group SO(3) is comprised of all 3×3 orthogonal matrices with a determinant of $+1$. Applying a matrix in SO(3) to a vector is equivalent to rotating that vector by some angle around a fixed axis. One way of describing the rotation is with Euler angles and Euler axes. An Euler angle is defined as the angle of rotation about a coordinate axis. By applying these angles of rotation in a specific order, one reference frame can be rotated into another. Euler's theorem states: "Any two independent orthonormal coordinate frames can be related by a sequence of rotations (not more than three) about coordinate axes, where no two successive rotations may be about the same axis" [69]. While there exist twelve different sequences of rotations, the sequence z-y-x is commonly used in aerospace applications.

In Figure 2.1 the Euler angles are labeled. A rotation about the z-axis is called yaw (ψ), a rotation about the y-axis is pitch (θ), and a rotation about the x-axis is roll (ϕ). Given the angles ϕ, θ, and ψ, any vector in one coordinate frame can be rotated into another coordinate frame. To rotate from the inertial frame to the body frame

$$\begin{aligned}
\mathbf{x}^B &= \begin{bmatrix} c_\psi c_\theta & s_\psi c_\theta & -s_\theta \\ c_\psi s_\theta s_\phi - s_\psi c_\phi & s_\psi s_\theta s_\phi + c_\psi c_\phi & c_\theta s_\phi \\ c_\psi s_\theta c_\phi + s_\psi s_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi & c_\theta c_\phi \end{bmatrix} \mathbf{x}^I \\
\mathbf{x}^B &= R_{IB}\mathbf{x}^I
\end{aligned}$$

where $c_x = \cos(x)$ and $s_x = \sin(x)$. R_{IB} is a rotation matrix, so it is orthogonal, and its inverse is its transpose. Therefore, to rotate from the body frame to the inertial frame $\mathbf{x}^I = R_{IB}^T\mathbf{x}^B = R_{BI}\mathbf{x}^B$. A difficulty arises with using Euler angles to represent rotations. No matter what rotation order is used, there is always a singularity at some angle.

A.2 Quaternions

A quaternion can be thought of as the composition of a scalar value and a three-vector:

$$q = q_0 + \mathbf{q} = q_0 + \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3.$$

Two quaternions p and q are equal if $p_i = q_i$ for $i = 0, 1, 2, 3$. The sum of two quaternions is defined as

$$p + q = (p_0 + q_0) + \mathbf{i}(p_1 + q_1) + \mathbf{j}(p_2 + q_2) + \mathbf{k}(p_3 + q_3).$$

The product (which is not commutative) of two quaternions is

$$pq = p_0q_0 - \mathbf{q} \cdot \mathbf{q} + p_0\mathbf{q} + q_0\mathbf{p} + \mathbf{p} \times \mathbf{q}.$$

The complex conjugate of a quaternion $q = q_0 + \mathbf{q}$ is $q^* = q_0 - \mathbf{q}$, and the norm of a quaternion is $|q| = \sqrt{q^*q} = \sqrt{qq^*}$. A quaternion's inverse is

$$q^{-1} = \frac{q^*}{||q||^2}.$$

The derivative of a quaternion is another quaternion. Suppose ω is the angular rate at which a rotation is occurring about some unit vector $\overline{\Omega} = \mathbf{i}\omega_1 + \mathbf{j}\omega_2 + \mathbf{k}\omega_3$. Then

$$\dot{q} = \omega q\overline{\Omega}$$

and

$$\dot{q}^* = -\omega\overline{\Omega}q^*.$$

These can be written in matrix form as

$$\dot{q} = \omega \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} q$$

and

$$\dot{q}^* = \omega \begin{bmatrix} 0 & \omega_1 & \omega_2 & \omega_3 \\ -\omega_1 & 0 & \omega_3 & -\omega_2 \\ -\omega_2 & -\omega_3 & 0 & \omega_1 \\ -\omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} q^*.$$

Quaternions are a way to represent rotations without singularities. To rotate a vector in $\mathbb{R}^3$, view the vector as a quaternion with its real part equal to zero (a pure quaternion). Then the rotation of a vector $\mathbf{v}$ via a quaternion q is defined as

$$\mathbf{w} = q\mathbf{v}q^*,$$

which is a linear operation. A series of rotations on vector $\mathbf{v}$ by quaternion p then q is defined as

$$\mathbf{w} = qp\mathbf{v}p^*q^*.$$

A unit quaternion may be written as

$$q = q_0 + \mathbf{q} = \cos(\theta) + \mathbf{u}\sin(\theta). \tag{A.1}$$

The rotation of vector $\mathbf{v}$ by a unit quaternion q as defined in Equation (A.1) may be interpreted as a rotation of $\mathbf{v}$ through an angle of 2θ about the axis $\mathbf{q}$. So to rotate some vector an angle of α about a vector $\mathbf{q}$, apply a rotation with the following quaternion

$$q = q_0 + \mathbf{q} = \cos(\frac{\alpha}{2}) + \mathbf{u}\sin(\frac{\alpha}{2}).$$

Appendix B

Spherical Mathematics

Since the earth is often modeled as a sphere, we present math associated with spheres that is used in modeling the earth's magnetic field. We begin by presenting spherical coordinates and a method to convert between Cartesian and spherical coordinates. Next, we present Legendre functions. We conclude the chapter by linking Legendre functions with Laplace's equation to yield the equations used to model the earth's magnetic field.

B.1 Spherical Coordinate System

A common coordinate system used in situations where points are located on a sphere is the spherical coordinate system. Points in the spherical coordinate system are represented by three values: a magnitude r, an azimuth angle ϕ, and an elevation angle θ. A point P in spherical coordinates in reference to Cartesian axes is shown in Figure B.1. Let P be located at (r, θ, ϕ) in spherical points. The variable $r \geq 0$ describes P's distance from the origin. The variable θ varies between 0 and π and describes the angle between the z-axis and some vector from the origin to P. ϕ varies between 0 and 2π. This variable describes the angle between the x-axis and the vector from the origin to the projection of point P in the xy plane.

Now let the same point P be located at (x, y, z) in Cartesian coordinates. Equations to convert between the spherical and Cartesian coordinates can now be developed. Since r is the distance between the origin and point P,

$$r = \sqrt{x^2 + y^2 + z^2}. \tag{B.1}$$

Then from trigonometry

$$\theta = \arccos\left(\frac{z}{r}\right), \tag{B.2}$$

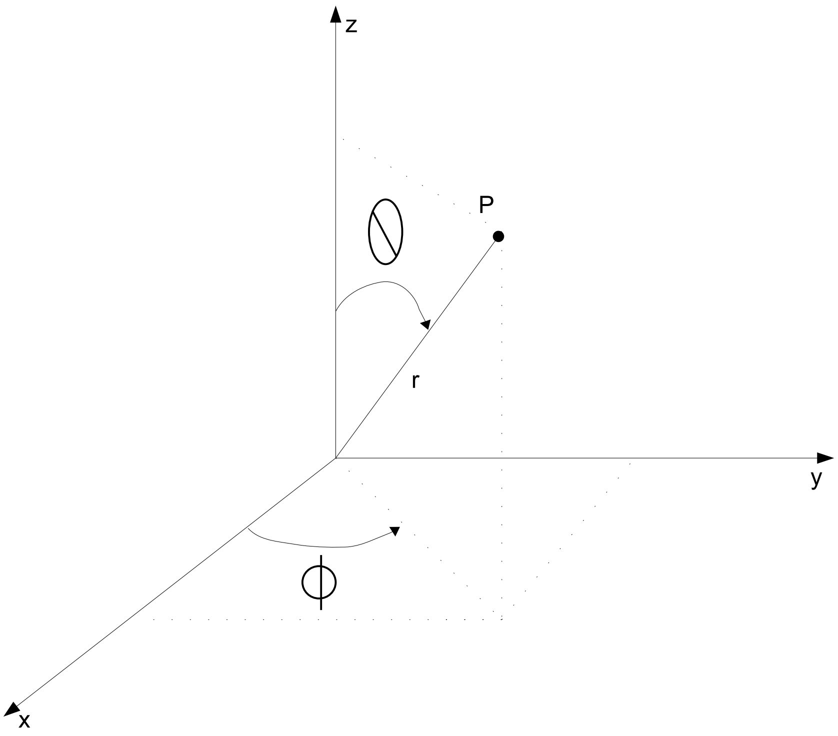

Figure B.1: The Spherical Coordinate System

and

$$\phi = \arctan\left(\frac{y}{x}\right). \tag{B.3}$$

By rearranging Equation (B.2)

$$z = r\cos(\theta). \tag{B.4}$$

From simple trigonometry the length of the projection of point P onto the xy plane is $r\cos(\frac{\pi}{2}-\theta) = r\sin(\theta)$. Then

$$x = r\sin(\theta)\cos(\phi), \tag{B.5}$$

and

$$y = r\sin(\theta)\sin(\phi). \tag{B.6}$$

B.2 Legendre Functions

Legendre functions are defined in terms of solutions to Legendre's differential equation

$$(1-z^2)\frac{d^2w}{dz^2}-2z\frac{dw}{dz}+[\nu(\nu+1)-\frac{\mu^2}{1-z^2}]w=0$$

where z, ν, and μ are complex variables [71]. The variables ν and μ are said to be the degree and order, respectively, of the system. Because the differential equation is second order, two linearly independent solutions exist for each z, ν, and μ. The regular solution at $z=\pm1$ is called a Legendre function of the first kind and shall be labeled $P_\nu^\mu(z)$. Likewise, the irregular solution at $z=\pm1$ is called a Legendre function of the second kind and shall be labeled as $Q_\nu^\mu(z)$. The solutions to Legendre's differential equation can be written equivalently in several forms.

It is common to use Legendre functions in the spherical coordinate system. Legendre functions of the first kind with integer degree and order have symmetry in spherical coordinates. Define a modified Legendre function as

$$S_n^m(x)=\sqrt{\frac{(n-m)!}{(n+m)!}}P_n^m(x) \qquad n\geq 0,\ n\geq m.$$

Then

$$\begin{aligned} P_{-n}^m(x) &= P_{n-1}^m(x), \\ P_n^{-m}(x) &= \frac{(n-m)!}{(n+m)!}P_n^m(x) \quad n\geq 0, \\ S_n^{-m}(x) &= S_n^m(x) \quad n\geq 0, \end{aligned}$$

and

$$P_n^m(-x)=(-1)^{n-m}P_n^m(x).$$

For a fixed n, the Legendre functions can be defined recursively:

$$\begin{aligned} P_n^m(x) &= [(2n-1)xP_{n-1}^m(x) - (n+m-1)P_{n-2}^m(x)]/(n-m) \\ S_n^m(x) &= [(2n-1)xS_{n-1}^m(x) - \sqrt{(n-1)^2 - m^2}S_{n-2}^m(x)]/\sqrt{n^2-m^2}. \end{aligned}$$

The recursion begins with

$$\begin{aligned} P_{m-1}^m(x) &= 0 \\ P_m^m(x) &= \frac{(2m)!(1-x^2)^{m/2}}{2^m m!} \\ S_{m-1}^m(x) &= 0 \\ S_m^m(x) &= \frac{\sqrt{(2m)!}(1-x^2)^{m/2}}{2^m m!}. \end{aligned}$$

B.3 Laplace's Equation

Laplace's equation is often encountered when studying and modeling the earth's magnetic field. In this section the solution of Laplace's equation in spherical coordinates is derived. The solution is then related to Legendre functions. The proceeding analysis is taken from [72].

B.3.1 Solution Derivation

In the spherical coordinate system, Laplace's equation is

$$\frac{\partial}{\partial r}\left(r^2\frac{\partial V}{\partial r}\right) + \frac{1}{\sin(\theta)}\frac{\partial}{\partial \theta}\left(\sin(\theta)\frac{\partial V}{\partial \theta}\right) + \frac{1}{\sin^2(\theta)}\left(\frac{\partial^2 V}{\partial \phi^2}\right) = 0. \tag{B.7}$$

Assume that the solution to Laplace's equation is variable-separable such that

$$V(r,\theta,\phi) = R(r)S(\theta\phi).$$

Then substituting the solution form into Equation (B.7)

$$\begin{aligned}
0 &= \frac{\partial}{\partial r}\left(r^2\frac{\partial}{\partial r}(RS)\right) + \frac{1}{\sin(\theta)}\frac{\partial}{\partial\theta}\left(\sin(\theta)\frac{\partial}{\partial\theta}(RS)\right) + \frac{1}{\sin^2(\theta)}\left(\frac{\partial^2}{\partial\phi^2}(RS)\right) \\
&= \frac{\partial}{\partial r}\left(r^2\left[R\cdot 0 + S\frac{\partial R}{\partial r}\right]\right) + \frac{1}{\sin(\theta)}\frac{\partial}{\partial\theta}\left(\sin(\theta)\left[R\frac{\partial S}{\partial\theta} + S\cdot 0\right]\right) + \\
&\quad \frac{1}{\sin^2(\theta)}\frac{\partial}{\partial\phi}\left[R\frac{\partial S}{\partial\phi} + S\cdot 0\right] \\
&= \frac{\partial}{\partial r}\left(Sr^2\frac{\partial R}{\partial r}\right) + \frac{1}{\sin(\theta)}\frac{\partial}{\partial\theta}\left(R\sin(\theta)\frac{\partial S}{\partial\theta}\right) + \frac{1}{\sin^2(\theta)}\frac{\partial}{\partial\phi}\left(R\frac{\partial S}{\partial\phi}\right) \\
&= S\frac{\partial}{\partial r}\left(r^2\frac{\partial R}{\partial r} + 0\cdot r^2\frac{\partial R}{\partial r}\right) + \frac{1}{\sin(\theta)}\left[R\frac{\partial}{\partial\theta}\left(\sin(\theta)\frac{\partial S}{\partial\theta}\right) + 0\cdot\sin(\theta)\frac{\partial S}{\partial\theta}\right] + \\
&\quad \frac{1}{\sin^2(\theta)}\left[R\frac{\partial^2 S}{\partial\phi^2} + 0\cdot\frac{\partial S}{\partial\phi}\right] \\
&= S\frac{\partial}{\partial r}\left(r^2\frac{\partial R}{\partial r}\right) + R\frac{1}{\sin(\theta)}\frac{\partial}{\partial\theta}\left(\sin(\theta)\frac{\partial S}{\partial\theta}\right) + R\frac{1}{\sin^2(\theta)}\frac{\partial^2 S}{\partial\phi^2}.
\end{aligned}$$

Divide by RS to yield

$$0 = \frac{1}{R}\frac{\partial}{\partial r}\left(r^2\frac{\partial R}{\partial r}\right) + \frac{1}{S\sin(\theta)}\frac{\partial}{\partial(\theta)}\left(\sin(\theta)\frac{\partial S}{\partial\theta}\right) + \frac{1}{S\sin^2(\theta)}\frac{\partial^2 S}{\partial\phi^2},$$

and rearrange

$$\frac{1}{R}\frac{d}{dr}\left(r^2\frac{dR}{dr}\right) = -(S\sin(\theta))^{-1}\frac{\partial}{\partial\theta}\left(\sin(\theta)\frac{\partial S}{\partial\theta}\right) - (S\sin^2(\theta))^{-1}\frac{\partial^2 S}{\partial\phi^2}. \tag{B.8}$$

The left-hand side of Equation (B.8) depends only on r, and the right-hand side of the equation depends only on θ and ϕ. Both sides are then equal to some constant, assigned value $n(n+1)$, since the left and right sides must be equal for all values of r, θ, and ϕ. The left-hand side of Equation (B.8) can be expressed as

$$\frac{d}{dr}\left(r^2\frac{dR}{dr}\right) - Rn(n+1) = 0,$$

which has solutions $R = r^n$ and $R = r^{-n-1}$. The right-hand side is written as

$$\frac{1}{\sin(\theta)}\frac{\partial}{\partial\theta}\left(\sin(\theta)\frac{\partial S}{\partial\theta}\right) + \sin^2(\theta)\frac{\partial^2 S}{\partial\phi^2} + Sn(n+1) = 0.$$

Assume now that the solution S is variable separable such that

$$S(\theta, \phi) = T(\theta)F(\phi).$$

Then

$$\frac{\sin(\theta)}{T}\frac{d}{d\theta}\left(\sin(\theta)\frac{dT}{d\theta}\right) + \sin^2(\theta)n(n+1) = -\frac{1}{F}\frac{d^2F}{d\phi^2}.$$

By the same logic as above, both sides of the equation are equal to some constant, m^2. Then

$$\frac{dF}{d\phi} + Fm^2 = 0$$

has solutions $F = \cos(m\phi)$ and $F = \sin(m\phi)$. The right-hand side is written as

$$\frac{d}{d\theta}\left(\sin(\theta)\frac{dT}{d\theta}\right) + T\left[n(n+1)\sin(\theta) - \frac{m^2}{\sin(\theta)}\right] = 0. \tag{B.9}$$

B.3.2 Relationship with Legendre Functions and Spherical Harmonics

If in Equation (B.9) $m = 0$, the equation becomes Legendre's equation

$$\frac{d}{dz}\left[(1-z^2)\frac{dT}{dz}\right] + n(n+1)T = 0 \tag{B.10}$$

where $z = \cos\theta$. Assume that the solution T is in power series form to yield

$$T = P_n(z) = \sum_{j=0}^{k}(-1)^j \frac{(2n-2j)!}{2^n j!(n-j)!(n-2j)!} z^{n-2j} \tag{B.11}$$

where k is an integer equal either to $\frac{1}{2}n$ or $\frac{1}{2}(n-1)$. $P_n(z)$ is known as a Legendre polynomial or a Legendre function of the first kind. The first few Legendre polynomials are

$$\begin{aligned}
P_0(z) &= 1 \\
P_1(z) &= z \\
P_2(z) &= \frac{1}{2}(3z^2-1) \\
P_3(z) &= \frac{1}{2}(5z^3-3z)
\end{aligned}$$

Solutions of Equation (B.9) are also called associated Legendre functions $P_n^m(\cos(\theta))$ since they can be derived by differentiating Equation (B.10) m times

$$P_n^m(\cos(\theta)) = N\sin^m(\theta)\frac{d^m}{d(\cos(\theta))^m}P_n\cos(\theta).$$

N is the Schmidt normalizing factor

$$N^2 = 2\frac{(n-m)!}{(n+m)!},$$

which makes the average value of $(P_n^m(\cos\theta))^2$ constant over the surface of a sphere for all m. The first few functions $P_n^m(\cos(\theta))$ are

	unnormalized	Schmidt normalized
$P_1^1(\cos(\theta))$	$\sin(\theta)$	$\sin(\theta)$
$P_2^1(\cos(\theta))$	$3\sin(\theta)\cos(\theta)$	$3^{-1/2}\sin(\theta)\cos(\theta)$
$P_2^2(\cos(\theta))$	$3\sin^2(\theta)$	$12^{-1/2}\sin^2(\theta)$

A solution of Laplace's equation can then be written as

$$V = (Ar^n + Br^{-n-1})(A'\cos(m\phi) + B'\sin(m\phi))P_n^m(\cos(\theta)) \quad m \leq n \tag{B.12}$$

where n and m are integers. The last two factors of Equation (B.12) can be combined for all possible values of m to give the surface spherical harmonic

$$S_n(\theta,\phi) = a_0 P_n(\cos(\theta)) + \sum_{m=1}^{n} (a_m \cos(m\phi) + b_m \sin(m\phi)) P_n^m(\cos(\theta)).$$

Surface spherical harmonics (derived from the Fourier series) are of the form

$$Y(\theta,\phi) = \sum_{n=0}^{\infty} \sum_{m=0}^{n} P_n^m(\cos(\theta))(a_n^m \cos(m\phi) + b_n^m \sin(m\phi))$$

where a_n^m and b_n^m are coefficients that can be used to define an arbitrary function (meeting some continuity conditions) on the surface of a sphere.

Appendix C

Kalman Filtering

Assume it is desired to know the states of the system

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} + B\mathbf{u} + \mathbf{w} & \text{(C.1)} \\ \mathbf{y} &= C\mathbf{x} + \mathbf{v} & \text{(C.2)} \end{aligned}$$

where $\mathbf{w}$ is zero mean Gaussian process noise and $\mathbf{v}$ is zero mean Gaussian measurement noise. Let $Q_c = E[\mathbf{w}\mathbf{w}^T]$, and $R = E[\mathbf{v}\mathbf{v}^T]$. $\mathbf{w}$ and $\mathbf{v}$ are also assumed to be uncorrelated. Further assume measurements of some of the states are unavailable. The Kalman filter is an optimal means of estimating states in the sense that it produces unbiased estimates ($E[\mathbf{e}] = 0$ where $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$) and minimizes the error covariance matrix ($P = E[\mathbf{e}\mathbf{e}^t]$).

C.1 Linear Systems

Since sensor measurements usually occur at regular intervals, the Kalman filter is most often performed on discrete systems. The system of Equations (C.1)-(C.2) discretized for some interval Δt is

$$\begin{aligned} \mathbf{x}_{k+1} &= A_d\mathbf{x}_k + B_d\mathbf{u}_k + \mathbf{w}_k & \text{(C.3)} \\ \mathbf{y}_k &= C_d\mathbf{x}_k + \mathbf{v}_k & \text{(C.4)} \end{aligned}$$

where $Q_d = E[\mathbf{w}_k\mathbf{w}_k^T]$, and $R_d = E[\mathbf{v}_k\mathbf{v}_k^T]$. The Kalman filter consists of two stages, the time update stage and the measurement update stage, which are performed in a continuous cycle. Time update can be considered a prediction stage that predicts states before measurements are considered. Measurement update can be considered a correction stage in which the predicted state from the time step are corrected with actual measurements.

The Kalman filter for linear systems is given in Algorithm C.1.1. A derivation of the equations can be found in [73]. A superscript minus sign in the algorithm denotes a time step variable while a superscript plus sign denotes a measurement update variable. If care is taken, the error covariance matrix P can be viewed as a measure of how good the filter's estimate is. As a word of caution when using the Kalman filter: In order for the Kalman filter's estimates to be reliable, the system must be observable.

```
Data: $\hat{\mathbf{x}}_0$ and $P_0$
Result: State estimate $\hat{\mathbf{x}}$ and error covariance matrix $P$
while Not done do
    Time Update;
    $\hat{\mathbf{x}}^-_{k+1} = A_d\hat{\mathbf{x}}^+_k + B_d\mathbf{u}_k$;
    $P^-_{k+1} = A_d P^+_k A_d^T + Q_d$;
    Gain calculation: $L_{k+1} = P^-_{k+1} C_d^T [C_d P^-_{k+1} C_d^T + R_d]^{-1}$;
    Measurement Update;
    $\hat{\mathbf{x}}^+_{k+1} = \hat{\mathbf{x}}^-_{k+1} + L_{k+1}(y - C_d\hat{\mathbf{x}}^-_{k+1})$;
    $P^+_{k+1} = (I - L_d C_d) P^-_{k+1}$;
end
```

Algorithm C.1.1: Kalman Filter

C.2 Extended Kalman Filter

Suppose a nonlinear system is defined by

$$\begin{aligned}
\dot{\mathbf{x}}(t) &= f(\mathbf{x},t) + g(\mathbf{x},t) + \mathbf{w}(t) \\
\mathbf{y}(t) &= h(\mathbf{x},t) + \nu(t)
\end{aligned}$$

where $\mathbf{w}$ and ν are zero-mean Gaussian noise with $E[\mathbf{w}\mathbf{w}^T] = Q$ and $E[\nu\nu^T] = R$. The Kalman filter is applied only to linear systems. The Extended Kalman Filter (EKF) is the estimator used for nonlinear systems. The EKF involves using the nonlinear equations in the prediction step and a linearized version of the system in the correction step. Before the EKF algorithm is begun, the Jacobian of $f(\mathbf{x}, t)$ and $h(\mathbf{x}, t)$ are found: $F = \frac{\partial f(\mathbf{x},t)}{\partial \mathbf{x}}$ and $H = \frac{\partial h(\mathbf{x},t)}{\partial \mathbf{x}}$. The discrete matrices F_d and H_d can be found by applying the matrix exponential: $F_d = e^{F\Delta t}$ and $H_d = e^{H\Delta t}$.

Once the preliminary steps are completed, the EKF algorithm can be performed. A time update is performed by using the nonlinear equations to predict the next state via numerical integration. An error covariance matrix is then computed from discrete matrices F_d and Q_d. The next step is the measurement update. A gain matrix L_k is calculated, and a corrected state estimate is produced. The complete method is given in Algorithm C.2.1.

```
Data: $\hat{\mathbf{x}}_0$ and $P_0$
Result: State estimate $\hat{\mathbf{x}}$ and error covariance matrix $P$
while Not done do
    Time Update;
    $\hat{\mathbf{x}}^-_{k+1} = \hat{\mathbf{x}}^+_k + \int_{t_k}^{t_{k+1}} f(\hat{x}(\tau), \tau) d\tau$ using a numerical integration technique such as
    Runge-Kutta or Euler;
    $P^-_{k+1} = F_d P^+_k F_d + Q_d$;
    Gain Calculation: $L_{k+1} = P^-_{k+1} H_d^T [H_d P^-_{k+1} H_d^T + R_k]^{-1}$;
    Measurement Update;
    $\hat{\mathbf{x}}^+_{k+1} = \hat{\mathbf{x}}^-_{k+1} + L_{k+1}(y - h(\hat{\mathbf{x}}^-_{k+1}, t))$;
    $P^+_{k+1} = (I - L_{k+1} H_d) P^-_{k+1}$;
end
```

Algorithm C.2.1: Extended Kalman Filter

C.3 Unscented Kalman Filter

Due to errors from linearization, the EKF sometimes fails to converge. The unscented Kalman filter (UKF), based on the unscented transform developed by Julier and Uhlmann [74], does not rely on linearization or partial derivatives [66]. Rather it constructs a set of points (sigma points) that have the same sample mean and covariance as the distribution of the random variable of interest. Each sigma point has an associated weight. Once the sigma points are constructed, they are propagated through the nonlinear transform to yield a set of transformed samples. These samples are then multiplied by their associated weights to yield a predicted mean, which in turn is used to compute a predicted covariance. The UKF is accurate to the third order for Gaussian distributions and requires the same order of computation as the EKF.

Let a nonlinear system be modeled as

$$
\begin{aligned}
\mathbf{x}_{k+1} &= F(\mathbf{u}_k, \mathbf{u}_k) + \mathbf{v}_k \\
\mathbf{y}_k &= H(\mathbf{x}_k) + \mathbf{n}_k
\end{aligned}
$$

where $\mathbf{x}$ is of order L, $\mathbf{v}_k$ is process noise with variance R^v, and $\mathbf{n}_k$ is sensor noise with variance R^n. To completely capture the variable's statistics, $2L+1$ sigma points are needed. The sigma points and associated weights as described in [75] are constructed as follows:

$$
\begin{aligned}
\chi_0 &= \overline{\mathbf{x}} \\
\chi_i &= \overline{\mathbf{x}} + (\sqrt{(L+\lambda)P_x})_i \quad i = 1, \ldots, L \\
\chi_i &= \overline{\mathbf{x}} - (\sqrt{(L+\lambda)P_x})_{i-L} \quad i = L-1, \ldots, 2L \\
W_0^{(m)} &= \lambda/(L+\lambda) \\
W_o^{(c)} &= \lambda/(L+\lambda) + (1-\alpha^2+\beta) \\
W_i^{(m)} &= W_i^{(c)} = 1/\{2(L+\lambda)\} \quad i = 1, \ldots, 2L,
\end{aligned}
$$

where $\lambda = \alpha^2(L+\kappa) - L$ is a scaling parameter, α determines the spread of the sigma points around $\overline{\mathbf{x}}$ (usually around 1e-3), κ is a scaling parameter usually set to zero, and β is used to incorporate prior knowledge of the distribution of $\mathbf{x}$ (for Gaussian distributions $\beta = 2$). The notation $(\sqrt{(L+\lambda)P_x})_i$ refers to the i^{th} row of the matrix square root. The UKF for state estimation as presented in [76] is shown in Algorithm C.3.1.

Van der Merwe and Wan developed a more computationally efficient implementation of the UKF called the square-root UKF (SRUKF)[76]. The SRUKF uses the QR decomposition, Cholesky factor updating, and efficient least-squares algorithms. The algorithm is presented in Algorithm C.3.2.

```
Initialize: $\hat{\mathbf{x}}_0 = E[x_0]$, $P_0 = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T]$;
for $k = 1, \ldots, \infty$ do
    Calculate Sigma Points:;
    $\chi_{k-1} = \left[\begin{array}{ccc} \hat{\mathbf{x}}_{k-1} & \hat{\mathbf{x}}_{k-1} + \eta\sqrt{P_{k-1}} & \hat{\mathbf{x}}_{k-1} - \eta\sqrt{P_{k-1}} \end{array}\right]$;
    Time Update;
    $\chi_{k|k-1} = F[\chi_{k-1}, \mathbf{u}_{k-1}]$;
    $\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \chi_{i,k|k-1}$;
    $P_k^- = \sum_{i=0}^{2L} W_i^{(c)} [\chi_{i,k|k-1} - \hat{\mathbf{x}}_k^-][\chi_{i,k|k-1} - \hat{\mathbf{x}}_k^-]^T + R^v$;
    $\psi_{k|k-1} = H[\chi_{k|k-1}]$;
    $\hat{\mathbf{y}}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \psi_{i,k|k-1}$;
    Measurement Update;
    $P_{\overline{y}_k \overline{y}_k} = \sum_{i=0}^{2L} W_i^{(c)} [\psi_{i,k|k-1} - \hat{\mathbf{y}}_k^-][\psi_{i,k|k-1} - \hat{\mathbf{y}}_k^-]^T + R^n$;
    $P_{\mathbf{x}_k \mathbf{y}_k} = \sum_{i=0}^{2L} W_i^{(c)} [\chi_{i,k|k-1} - \hat{\mathbf{x}}_k^-][\psi_{i,k|k-1} - \hat{\mathbf{y}}_k^-]^T$;
    $K = P_{\mathbf{x}_k \mathbf{y}_k} P_{\overline{\mathbf{y}}_k \overline{\mathbf{y}}_k}^{-1}$;
    $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K(\mathbf{y}_k - \hat{\mathbf{y}}_k^-)$;
    $P_k = P_k^- - K_k P_{\mathbf{x}_k \mathbf{y}_k} K_k^T$;
end
where $\eta = \sqrt{(L + \lambda)}$.;
```

Algorithm C.3.1: UKF for State Estimation

```
Initialize: $\hat{\mathbf{x}}_0 = E[x_0]$, $S_0 = \text{chol}\{E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T]\}$;
for $k = 1, \ldots, \infty$ do
    Calculate Sigma Points:;
    $\chi_{k-1} = \left[\ \hat{\mathbf{x}}_{k-1} \quad \hat{\mathbf{x}}_{k-1} + \eta S_k \quad \hat{\mathbf{x}}_{k-1} - \eta S_k\ \right]$;
    Time Update;
    $\chi_{k|k-1} = F[\chi_{k-1}, \mathbf{u}_{k-1}]$;
    $\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \chi_{i,k|k-1}$;
    $S_k^- = \text{qr}\{\left[\ \sqrt{W_1^{(c)}}(\chi_{1:2L,k|k-1} - \hat{\mathbf{x}}_k^-) \quad \sqrt{R^v}\ \right]\}$;
    $S_k^- = \text{cholupdate}\{S_k^-, \chi_{0,k} - \hat{\mathbf{x}}_k^-, W_0^{(c)}\}$ $\psi_{k|k-1} = H[\chi_{k|k-1}]$;
    $\hat{\mathbf{y}}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \psi_{i,k|k-1}$;
    Measurement Update;
    $S_{\overline{\mathbf{y}}_k}^- = \text{qr}\{\left[\ \sqrt{W_1^{(c)}}(\psi_{1:2L,k|k-1} - \hat{\mathbf{y}}_k) \quad \sqrt{R_k^n}\ \right]\}$;
    $S_{\overline{\mathbf{y}}_k}^- = \text{cholupdate}\{S_{\overline{\mathbf{y}}_k}^-, \psi_{0,k} - \hat{\mathbf{y}}_k, W_0^{(c)}\}$
    $P_{\mathbf{x}_k\mathbf{y}_k} = \sum_{i=0}^{2L} W_i^{(c)}[\chi_{i,k|k-1} - \hat{\mathbf{x}}_k^-][\psi_{i,k|k-1} - \hat{\mathbf{y}}_k^-]^T$;
    $K_k = (P_{\mathbf{x}_k\mathbf{y}_k}/S_{\overline{\mathbf{y}}_k}^T)/S_{\overline{\mathbf{y}}_k}$;
    $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K(\mathbf{y}_k - \hat{\mathbf{y}}_k^-)$;
    $U = K_k S_{\overline{\mathbf{y}}_k}$;
    $S_k = \text{cholupdate}\{S_k^-, U, -1\}$;
end
where $\eta = \sqrt{(L + \lambda)}$.;
```

Algorithm C.3.2: Square Root Implementation of the UKF

Appendix D

Latitude and Longitude

Latitude and longitude are a convenient way to describe position on a spherical surface such as the Earth. Latitude and longitude are typically divided into units of degrees, minutes, and seconds where a degree contains 60 minutes, and a minute contains 60 seconds. Lines of latitude run from $+90°$ (or 90° N) at the North Pole to $-90°$ (or 90° S) at the South Pole with the equator being at 0°. Lines of longitude vary from 180° (or 90° E) to $-180°$ (or 180° W) with 0° being the Prime Meridian which runs through Greenwich, England.

D.1 Cartesian Calculations

In this section we summarize a method found in [77] to calculate latitude and longitude in spherical geocentric coordinates. To use the WMM, it is necessary to have a latitude and a longitude. So given an initial latitude and longitude as well as a distance traveled in the northerly and easterly directions, it is necessary to calculate the new latitude and longitude. Spherical geometry can be used to do this if the Earth is modeled as a sphere. The Earth is not actually spherical as it bulges at the equator.

Several points are labeled in Figure D.1 that aid in finding the new latitude and longitude. Let point A be the starting point with a known initial latitude $lat1$ and longitude $lon2$, and let point B be the ending point located at latitude $lat2$ and longitude $lon2$ after traveling a distance of X in the easterly direction and Y in the northerly direction. Point D is located at the same latitude as A and the same longitude as point B. Likewise, point C is located at the same longitude as A and the same latitude as B. Point E is the point where A's longitude intersects the equator, and point F is where B's longitude intersects the equator. Point O is located at the Earth's center.

The derivation of $lon2$ is straightforward. To find $lon2$ first find the angle $\angle AOE$ which is equal to $lat1$ by the definition of latitude. Next, the radius of the circle that passes through points

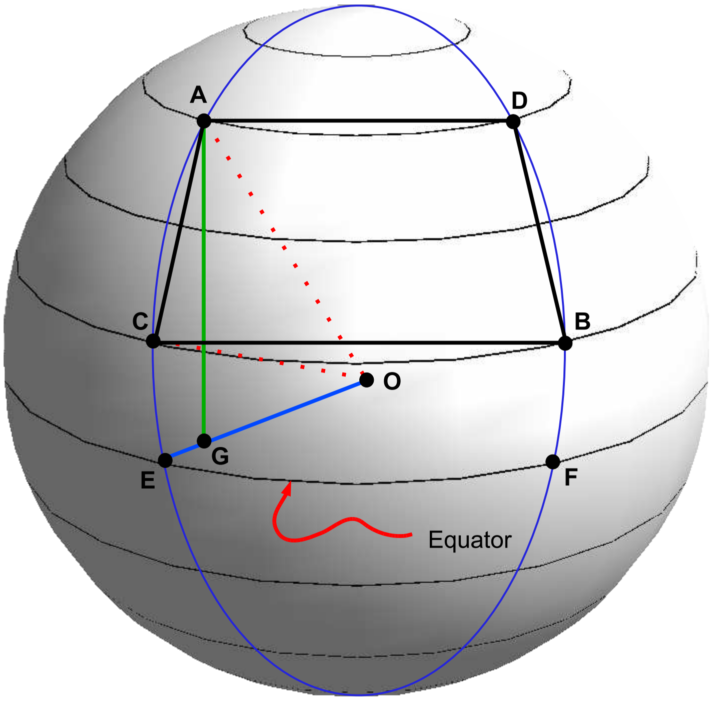

Figure D.1: Derivation of Latitudes and Longitudes

A and D at $lat1$ is needed. This is found by first dropping a perpendicular from A to the line segment OE. The perpendicular intersects OE at point G. Then the length of segment OG is equal to the desired radius. From trigonometry the length of OG is

$$OG = R\cos(lat1)$$

where R is the Earth's radius. Let O' be the center of the circle that forms $lat1$. Then the angle $\angle AO'D$ is given by

$$\angle AO'D = \frac{Y}{R\cos(lat1)}$$

since Y is the arc length from A to D and $R\cos(lat1)$ is the circle's radius. Then $lon2$ is given by

$$\begin{aligned} lon2 &= lon1 + \angle AO'D \\ &= lon1 + \frac{Y}{R\cos(lat1)}. \end{aligned} \tag{D.1}$$

The derivation for $lat2$ is also relatively simple. First note that the angle $\angle AOC$ is

$$\angle AOC = lat2 - lat1$$

by the definition of latitude. Then the arc length X between points A and C is

$$\begin{aligned} X &= R\angle AOC \\ &= R(lat2 - lat1). \end{aligned}$$

Then $lat2$ is given by

$$lat2 = lat1 + \frac{X}{R}. \tag{D.2}$$

D.2 Geodetic Coordinates

In this section we summarize the geodetic reference frame as presented in [78]. For geodetic coordinates the Earth is modeled as an ellipsoid as shown in Figure D.2. The x-axis is in the equatorial plane and fixed with the Earth's rotation such that it passes through the Greenwich meridian. The z-axis is parallel to the Earth's rotation axis, and the y-axis is perpendicular to the x- and z-axes according to the right-hand rule. A cross-section is taken such that the major axis of an ellipse corresponds to the equator. The center of the Earth is at point O. Then the semimajor axis a has a value equal to the Earth's mean radius, and the semiminor axis b is equal to the Earth's polar diameter. With values a and b determined by which physical model of the Earth is used, the

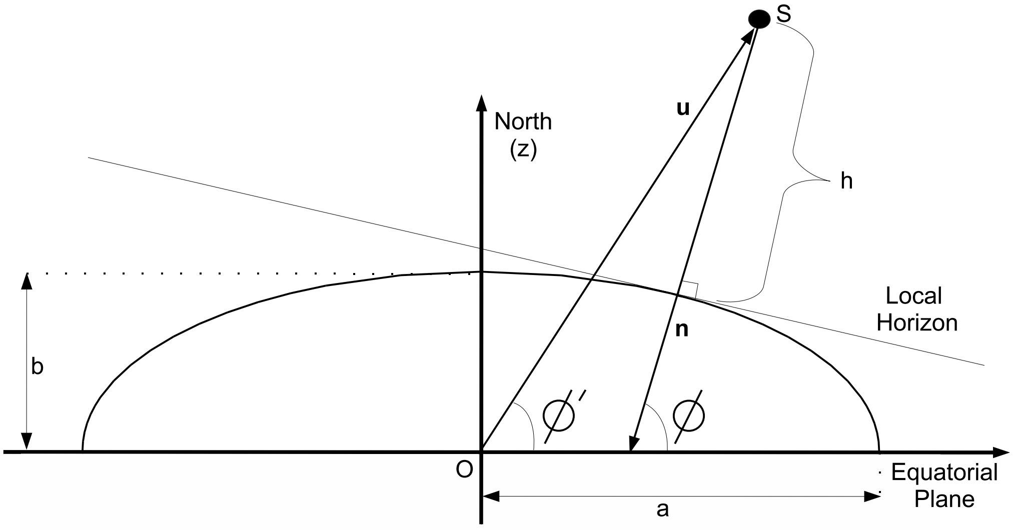

Figure D.2: Geodetic Coordinate System

eccentricity e and the flattening f of the ellipsoid are

$$\begin{aligned} e &= \sqrt{1-\frac{b^2}{a^2}} \\ f &= 1-\frac{b}{a}. \end{aligned}$$

Geodetic latitude, longitude, and height are defined with respect to the reference ellipsoid. In Figure D.2 the vehicle's position is given by point S, which is defined by vector $\mathbf{u} = (x_u, y_u, z_u)$ in geocentric coordinates. Then the geodetic longitude λ, defined as the angle between the vehicle and the x-axis, measured in the xy-plane is

$$\lambda = \begin{cases} \tan^{-1}\left(\frac{y_u}{x_u}\right), & x_u \geq 0 \\ \frac{\pi}{2} + \tan^{-1}\left(\frac{y_u}{x_u}\right), & x_u < 0, y_u \geq 0 \\ -\frac{\pi}{2} + \tan^{-1}\left(\frac{y_u}{x_u}\right), & x_u < 0, y_u < 0. \end{cases}$$

Geodetic latitude ϕ is the angle between the ellipsoid normal vector $\mathbf{n}$ and the projection of $\mathbf{n}$ onto the xy-plane. Geocentric latitude is ϕ'. Geodetic height h is the minimum distance from the vehicle to the ellipsoid. To convert from geodetic to geocentric coordinates use

$$\begin{aligned}
x_u &= \frac{a\cos(\lambda)}{\sqrt{1+(1-e^2)\tan^2(\phi)}} + h\cos(\lambda)\cos(\phi) \\
y_u &= \frac{a\sin(\lambda)}{\sqrt{1+(1-e^2)\tan^2(\phi)}} + h\sin(\lambda)\cos(\phi) \\
z_u &= \frac{a(1-e^2)\sin(\phi)}{\sqrt{1-e^2\sin^2(\phi)}} + h\sin(\phi).
\end{aligned}$$