UNIVERSITY OF CALIFORNIA, SAN DIEGO

Quaternion-Based Aircraft Attitude Estimation

A thesis submitted in partial satisfaction of the
requirements for the degree Master of Science

in

Engineering Sciences (Aerospace Engineering)

by

Brian Michael Filarsky

Committee in charge:

Mauricio de Oliveira, Chair
Jorge Cortes, Co-Chair
Mark Anderson

2016

The Thesis of Brian Michael Filarsky is approved and it is acceptable in quality and form for publication on microfilm and electronically:

Co-Chair

Chair

University of California, San Diego

2016

DEDICATION

This Thesis is dedicated to Mom, Dad, Sister, and Uncle Steve, who all encouraged me to pursue a graduate degree. To my mom and dad, for encouraging an interest in science and mathematics from a young age, and for patiently listening to me passionately explaining whatever problem of the week I had recently figured out. To my sister, Lauren, for all of her help in reading, editing, and shaking her head every time that I used "which" instead of "that". To Uncle Steve for the encouragement to make the extra push to get into graduate school.

I would like to thank my advisor, Professor de Oliveira, for his guidance and patience, and for helping me to see the light.

TABLE OF CONTENTS

LIST OF SYMBOLS

Type	**Desciption**	**Example**
matrix	uppercase letter	M
rotation matrix	C with subscript indicating frame	$C_{a \to b}$
rotation matrix from quaternion	matrix of a quaternion	$C_{a \to b}(\bar{q}_{a \to b})$
scalar	lowercase letter	s
imaginary number	italicized bold lowercase letter	$\boldsymbol{i}$
vector	non-italicized bold lowercase	$\mathbf{v}$
quaternion	lowercase letter with overbar	$\bar{q}$
quaternion multiplication	circle ×	$\otimes$
pure quaternion	vector with overbar	$\bar{\mathbf{v}}$
rotation quaternion	quaternion with subscript indicating frame from → to	$\bar{q}_{a \to b}$
rotation quaternion from matrix	quaternion of matrix	$\bar{q}_{a \to b}(C_{a \to b})$
time derivative	dot	$\dot{x}$
measurement	tilde accent	$\tilde{x}$
estimate	hat accent	$\hat{x}$
reduced representation	check accent	$\check{x}$
frame	superscipt	x^a
reference	subscript	x_r
error	delta	Δx
timestep	delta subscript t	Δ_t

LIST OF FIGURES

ABSTRACT OF THE THESIS

Quaternion-Based Aircraft
Attitude Estimation

by

Brian Michael Filarsky

Master of Science Engineering Sciences (Aerospace Engineering)

University of California, San Diego, 2016

Mauricio de Oliveira, Chair
Jorge Cortes, Co-Chair

Aircraft attitude estimation requires fusing several sensors in order to recover both high and low frequency information in an observable manner. This thesis explores the fusion of gyroscope integration, gravity vector estimation, and magnetic field vector estimation using a complementary filter and an extended Kalman filter (EKF), both of which use a unit quaternion to represent the attitude portion of the state.

First, a set of models, which contain bias, scale factor errors, alignment errors, and Gaussian white noise, is introduced to govern the available sensors. The gyroscope bias is modeled as a random walk. A calibration routine is then established to minimize scale factor and bias errors. After some definitions and derivations for quaternion algebra are established, the attitude solution is then estimated using the complementary filter. Then the EKF is introduced and used to estimate both the quaternion state and gyroscope bias.

The thesis is concluded with a Monte Carlo run to compare the complementary filter with the EKF. Due in large part to the estimation of gyroscope bias in the EKF, this filter is shown to give a significantly more accurate state estimate. The robustness is also evaluated, with both filters initialized with the incorrect initial quaternion and gyroscope bias estimate. The EKF is shown to converge relatively quickly, while the complementary filter does not reliably converge due to the lack of gyroscope bias estimation.

Introduction

Strap-down airborne navigation systems are everywhere. From spacecraft, commercial airliners, and military hardware using high-cost, high-precision components, to light aircraft using mid-range components, to personal drones using low-end components, the need to fuse measurements from multiple sensors into an attitude estimate is ubiquitous. This thesis looks mainly to derive, understand, and evaluate methods to fuse low-cost sensors for use in an airplane.

The attitude of an air vehicle can be represented in multiple forms. The most familiar and intuitive representation is the Tait-Bryan angles of heading, pitch, and roll. The most familiar to engineers is the rotation matrix, and a very useful but less intuitive is the unit quaternion. Each of the three methods mentioned have their own benefits and drawbacks. Tait-Bryan angles are very intuitive to visualize and are the way pilots refer to and think about aircraft attitude; however, they are mathematically and numerically challenging. They require rigorous use of transcendental functions and are prone to gimbal-lock; whenever the attitude is straight up or straight down, roll and heading become undefined. Neither rotation matrices nor quaternions suffer from this phenomenon. Rotation matrices are bound by significant constraints (must be orthonormal), which are difficult to maintain, and they require nine elements to represent a rotation. Rotating either a vector or a rotation matrix by another rotation matrix, however, is quite simple. Quaternions have a unit constraint, which does introduce some issues, but it is easier to handle than the orthonormal constraint on the rotation matrix. Quaternions are also represented by only four elements, and they can be used to rotate other quaternions with little effort. While quaternions can be used directly to rotate a vector, it will be shown to be equivalent to generating a rotation matrix, then rotating the vector by that matrix. For simplicity, the latter method will be used.

Due to the benefits and drawbacks of the three aforementioned methods, this thesis uses each of them where best suited. The plant and estimate in all of the filters are tracked in quaternions; rotation matrices are used to rotate vectors; and Tait-Bryan angles are generally used for plots.

Chapter 1

Modeling Sensors

Prior to attempting to estimate the vehicle position or attitude, a realistic model for the various sensors in use is desired. The sensors include a 3-axis rate gyroscope, a 3-axis accelerometer, and a 3-axis magnetometer. First, the equations governing the data the sensors capture need to be derived. Starting with body-frame velocity, acceleration, rotation rate, and a given magnetic field, the following equations are used to model the sensors.

The gyroscope is the simplest IMU component, as it directly measures roll, pitch, and yaw rate in the body frame, which is an input to the simulation.

$$\boldsymbol{\omega}^b = \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{1.1}$$

The Earth's magnetic field has some dip angle i, which is the angle between the horizontal and the field lines, with positive being down. There is also a declination angle δ, which is the angle between true north and the magnetic field, with positive being to the east. Taking m as the strength of the magnetic field, the field in body coordinates is [1]:

$$\mathbf{m}^b = C_{n\to b} C_{m\to n} \begin{pmatrix} m \\ 0 \\ 0 \end{pmatrix} \tag{1.2}$$

The accelerometer measures the force applied to it, which is body acceleration minus the body gravity vector. To derive the equations for body acceleration [1], first the mapping between body and local velocity must be completed:

$$\mathbf{v}^n = C_{b\to n}\mathbf{v}^b \tag{1.3}$$

Taking the derivative with respect to time:

$$\dot{\mathbf{v}}^n = C_{b\to n}\dot{\mathbf{v}}^b + \dot{C}_{b\to n}\mathbf{v}^b \tag{1.4}$$

Then multiplying from the left by $C_{n\to b}$:

$$C_{n\to b}\dot{\mathbf{v}}^n = \dot{\mathbf{v}}^b + C_{n\to b}\dot{C}_{b\to n}\mathbf{v}^b \tag{1.5}$$

The skew-symmetric matrix Ω is defined:

$$C_{n\to b}\dot{C}_{b\to n} = \Omega = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \tag{1.6}$$

Finally, the body acceleration of the vehicle:

$$\mathbf{a}^b = \dot{\mathbf{v}}^b + \Omega\mathbf{v}^b \tag{1.7}$$

An accelerometer doesn't directly measure acceleration; it measures the force per unit mass applied to it, or specific force. Defining the gravity vector in the body frame:

$$\mathbf{g}^b = C_{n\to b} \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \tag{1.8}$$

The specific force is acceleration minus the gravity vector:

$$\mathbf{f}^b = \mathbf{a}^b - \mathbf{g}^b \tag{1.9}$$

The magnetometer, accelerometer, and gyroscope are modeled with bias, scale factor errors, alignment errors, and additive white Gaussian noise (AWGN). Additionally, the gyroscope is modeled with acceleration errors and a bias drift. The matrix S accounts for scale factor errors and alignment errors, with diagonal terms indicating scale factor errors and off-diagonal terms indicating alignment errors:

$$S = \begin{bmatrix} s_{xx} & m_{xy} & m_{xz} \\ m_{yx} & s_{yy} & m_{yz} \\ m_{zx} & m_{zy} & s_{zz} \end{bmatrix} = I + \begin{bmatrix} \eta_1 & \eta_2 & \eta_2 \\ \eta_2 & \eta_1 & \eta_2 \\ \eta_2 & \eta_2 & \eta_1 \end{bmatrix} \tag{1.10}$$

S is modeled as an identity matrix with error terms η_1 on the diagonal and η_2 off the diagonal. η_1 and η_2 are Gaussian with standard deviation depending on the sensor. The standard deviation of η_1 is larger than that of η_2 for all sensor models. Similar to S, the matrix A is used for the acceleration errors on the gyroscope, but with equal distribuitions on all elements and no identity matrix:

$$A = \begin{bmatrix} a_{xx} & a_{xy} & a_{xz} \\ a_{yx} & a_{yy} & a_{yz} \\ a_{zx} & a_{zy} & a_{zz} \end{bmatrix} = \begin{bmatrix} \eta_3 & \eta_3 & \eta_3 \\ \eta_3 & \eta_3 & \eta_3 \\ \eta_3 & \eta_3 & \eta_3 \end{bmatrix} \tag{1.11}$$

The $\mathbf{b}$ vector is sensor bias (also pulled from a Gaussian distribution), and $\mathbf{w}$ is zero-mean AWGN that is uncorrelated with any other signals.[1] The accelerometer model is then:

$$\tilde{\mathbf{a}} = S_a\mathbf{f}^b + \mathbf{b}_a + \mathbf{w}_a \tag{1.12}$$

The magnetometer model is:

$$\tilde{\mathbf{m}} = S_m\mathbf{m}^b + \mathbf{b}_m + \mathbf{w}_m \tag{1.13}$$

[1] The standard deviation of all sensor errors is included in the appendix.

And the gyroscope model is:

$$\tilde{\boldsymbol{\omega}} = S_\omega \boldsymbol{\omega}^b + A\mathbf{f}^b + \mathbf{b}_\omega + \mathbf{w}_\omega \tag{1.14}$$

The gyroscope model differs from the accelerometer and magnetometer in that it is affected by specific force, and that the bias drifts over time as a random walk:

$$\dot{\mathbf{b}}_\omega = \mathbf{w}_{\omega d} \tag{1.15}$$

While inertial velocity is also a required measurement, integration of the IMU components is the main focus of this thesis, and the modeling and estimation required to properly implement inertial velocity measurements was considered beyond scope. For this reason, the velocity measurement used is a simplified model using the plant with AWGN.

$$\tilde{\mathbf{v}} = \mathbf{v} + \mathbf{w}_v \tag{1.16}$$

All of the errors are pulled from a Gaussian distribution pseudo-random number generator, and the standard distributions of these are varied for each type of noise to try to give results in line with low-cost sensors. The uncalibrated outputs of the sensor models in 6 different attitudes are shown in figures 1.1, 1.2 and 1.3.

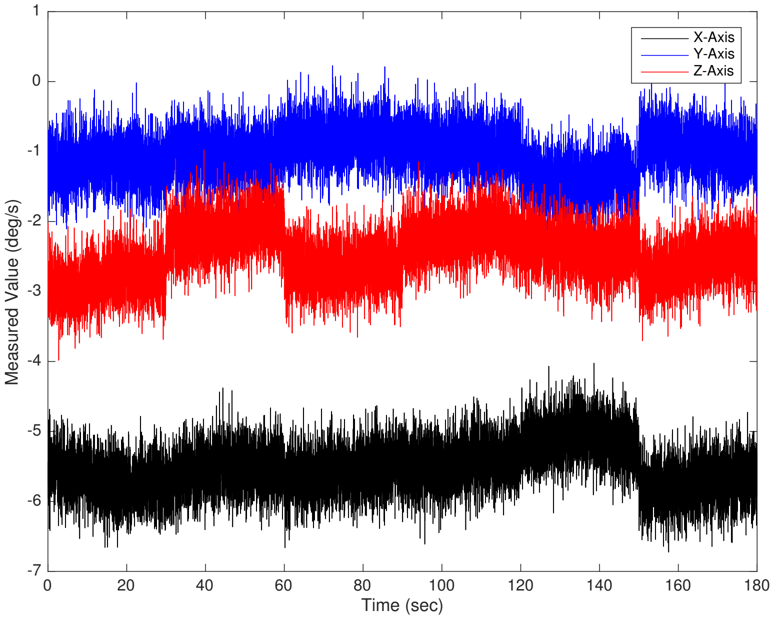

Figure 1.1: Uncalibrated gyroscope output in 6 sensor orientations

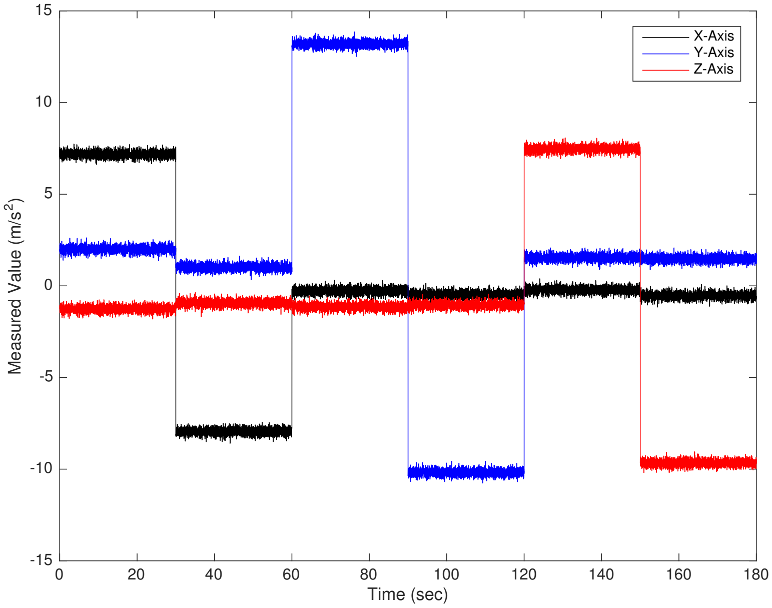

Figure 1.2: Uncalibrated accelerometer output in 6 sensor orientations

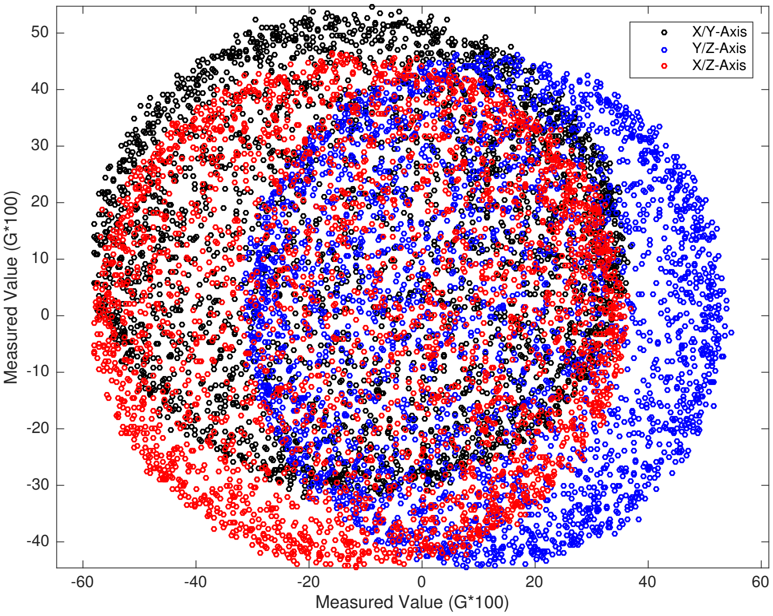

Figure 1.3: Uncalibrated magnetometer output of random rotations

Chapter 2

Calibrating Sensors

The accelerometer and gyroscope are run for thirty seconds in each of six orientations: each axis is aligned directly towards and directly away from the gravity vector. This allows for an estimation of the bias, scale factor, and alignment errors.

For the accelerometer, since the noise is zero-mean AWGN, $E(\mathbf{w}_a) = 0$, and acceleration is 0, the mean of an equal number of samples for the x-axis aligned with and away from the gravity vector is:

$$\overbrace{g \begin{pmatrix} s_{xx} \\ m_{yx} \\ m_{zx} \end{pmatrix} + \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix}}^{\text{Samples pointing towards axis}} \quad \overbrace{-g \begin{pmatrix} s_{xx} \\ m_{yx} \\ m_{zx} \end{pmatrix} + \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix}}^{\text{Samples pointing away from axis}} \quad = 2 \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} \tag{2.1}$$

This is also the case with the other axes. Bias errors for each axis are calculated by taking the mean of all of the samples for each axis (given the number of +/- samples for each axis are equal). Scale factor and alignment errors are found by subtracting the mean of the samples facing away from the gravity vector from the mean of the samples aligned with the gravity vector. For the x-axis, this is:

$$g\begin{pmatrix} s_{xx} \\ m_{yx} \\ m_{zx} \end{pmatrix} + \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} - \left(-g\begin{pmatrix} s_{xx} \\ m_{yx} \\ m_{zx} \end{pmatrix} + \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} \right) = 2g\begin{pmatrix} s_{xx} \\ m_{yx} \\ m_{zx} \end{pmatrix} \tag{2.2}$$

This isolates the scale and alignment errors for each axis. The procedure is repeated for each axis, until an estimate of S_a is found. With estimates of both S_a and $\mathbf{b}_a$ in hand (using $\hat{S}_a$ and $\hat{\mathbf{b}}_a$ as the estimate), they can then be applied to the measurements for calibrated samples:

$$\tilde{\mathbf{f}}_{cal} = \hat{S}_a^{-1}(\tilde{\mathbf{f}} - \hat{\mathbf{b}}_a) \tag{2.3}$$

The gyroscope is calibrated in the same way as the accelerometer, but since the bias drifts over time (and is allowed to drift for 5 seconds between each calibration orientation), the corrections are not quite as accurate. When adding the sum of the results in each opposing orientation, bias is once again recovered. When subtracting the sum, the terms remaining are for A. Without the use of a calibrated turntable, there is no way to calibrate for S_ω. The calibrated gyroscope output is then given by:

$$\tilde{\boldsymbol{\omega}}_{cal} = \tilde{\boldsymbol{\omega}} - \hat{A}\tilde{\mathbf{f}}_{cal} - \hat{\mathbf{b}}_\omega \tag{2.4}$$

The magnetometer calibration poses some challenges. While determining the angle of the magnetic field vector then taking measurements with each axis pointed to and away from it is possible, it can certainly be a bit more of a challenge than aligning the other sensors with the gravity vector. A more hassle-free calibration routine is desired for the magnetometer.

The data collect for the magnetometer consists of moving the sensor randomly in every direction in order to fill a sphere of magnetic vectors as fully as possible, as shown in figure 1.3. The challenge then is to determine the scale error and bias, as the true orientation of the magnetometer with respect to the magnetic field is not known for any of the samples.

The simplest method consists of finding the maximum and minimum values on each axis, then determining the scale and bias in a similar treatment to the other sensors. This has a number of drawbacks: the calibration only ends up using six samples (2 per axis); the maximum and minimum values will be the values affected by the maximum and minimum noise level, respectively; and it relies on a set of magnetic field vectors with sufficient coverage pointing towards and away from the magnetic field on each axis.

In order to accomplish this, first the maximum and minimum values of each axis are found. As there is AWGN included in the output, the absolute value of the maximum and minimum value will be inflated. As the maximum value on each axis will approach 0 on the other 2 axes, the off-diagonal terms in the scale matrix can be neglected. This gives:

$$\begin{pmatrix} \tilde{m}_x \\ \tilde{m}_y \\ \tilde{m}_z \end{pmatrix}_{max} = \begin{pmatrix} ms_{xx} \\ ms_{yy} \\ ms_{zz} \end{pmatrix} + \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} + \begin{pmatrix} \eta_x \\ \eta_y \\ \eta_z \end{pmatrix} \tag{2.5}$$

$$\begin{pmatrix} \tilde{m}_x \\ \tilde{m}_y \\ \tilde{m}_z \end{pmatrix}_{min} = \begin{pmatrix} -ms_{xx} \\ -ms_{yy} \\ -ms_{zz} \end{pmatrix} + \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} - \begin{pmatrix} \eta_x \\ \eta_y \\ \eta_z \end{pmatrix} \tag{2.6}$$

Adding (2.5) to (2.6) reveals the bias, as with the accelerometer and gyroscope, but subtracting (2.6) from (2.5) leads to an overestimate of the scale factor error (using +/- superscripts to indicate maximum and minimum readings):

$$\begin{pmatrix} \tilde{m}_x^+ - \tilde{m}_x^- \\ \tilde{m}_y^+ - \tilde{m}_y^- \\ \tilde{m}_z^+ - \tilde{m}_z^- \end{pmatrix} = 2 \begin{pmatrix} ms_{xx} \\ ms_{yy} \\ ms_{zz} \end{pmatrix} + 2 \begin{pmatrix} \eta_x \\ \eta_y \\ \eta_z \end{pmatrix} \tag{2.7}$$

After finding the standard deviation of the signal noise with the sensors at rest, assuming $\eta = 2\sigma$ and subtracting from the above equation leads to an estimate of scale matrix, $\hat{S}_m$.

While this approach works fairly well in simulation, it leaves much to be desired. The true efficacy on real hardware is questionable, as the data collection needs to be near-perfect, and a single "bad" measurement outside that expected by a Gaussian distribution could significantly impact the accuracy of the calibration.

A significantly more robust approach is to find a best fit for all the calibration data. Approaching it in this manner removes all of the issues associated with the first attempt: the fit is applied to all the calibration samples, the effect of the noise will tend to cancel out, and the accuracy is not dependent on a near-perfect set of vectors. Using a Gauss-Newton algorithm to find the least squares fit to an off-center tri-axial ellipsoid was determined to be a much better candidate for calibration.

The initial shape of the measured magnetic field vectors will be a tri-axial ellipsoid; the semi-principle axes are the scale factor error, and the center is the bias. By determining these parameters of the ellipsoid, bias and scale factor error are recovered, and using these to calibrate the sensor effectively shifts and scales the ellipsoid to be a sphere centered at the origin. The equation for the off-center tri-axial ellipsoid is:

$$\frac{(x-b_x)^2}{{s_{xx}}^2} + \frac{(y-b_y)^2}{{s_{yy}}^2} + \frac{(z-b_z)^2}{{s_{zz}}^2} = 1 \tag{2.8}$$

For some multi-variate non-linear function $f(x, \beta)$, with measured inputs x_i and residuals r_i, the Gauss-Newton least squares solution to parameters β is [2]:

$$r_i(\beta_{k-1}) = y_i - f(x_i, \beta_{k-1}) \tag{2.9}$$

$$J_{ij} = \frac{\partial r_i(\beta_{k-1})}{\partial \beta_j} \tag{2.10}$$

$$\beta_k = \beta_{k-1} - (J^T J)^{-1} J^T r(\beta_{k-1}) \tag{2.11}$$

Taking $\beta = \begin{pmatrix} b_x & b_y & b_z & s_{xx} & s_{yy} & s_{zz} \end{pmatrix}^T$ each row of the residuals and the

Jacobian is respectively:

$$r_i(\beta_{k-1}) = 1 - \frac{(x_i - b_x)^2}{{s_{xx}}^2} - \frac{(y_i - b_y)^2}{{s_{yy}}^2} - \frac{(z_i - b_z)^2}{{s_{zz}}^2} \tag{2.12}$$

$$J_i = \begin{pmatrix} \frac{2(x_i - b_x)}{s_{xx}^2} & \frac{2(y_i - b_y)}{s_{yy}^2} & \frac{2(z_i - b_z)}{s_{zz}^2} & \frac{(x_i - b_x)^2}{s_{xx}^3} & \frac{(y_i - b_y)^2}{s_{yy}^3} & \frac{(z_i - b_z)^2}{s_{zz}^3} \end{pmatrix} \tag{2.13}$$

Upon inspection, clearly the right three columns are linearly dependent on the left three columns. As $J^T J$ needs to be inverted, this will not suffice. However, multiplying each term in (2.8) by the denominator and inserting into the framework of (2.9) gives a Jacobian that is full column rank:

$$r_i(\beta_{k-1}) = s_{xx}^2 s_{yy}^2 s_{zz}^2 - (x_i - b_x)^2 s_{yy}^2 s_{zz}^2 - (y_i - b_y)^2 s_{xx}^2 s_{zz}^2 - (z_i - b_z)^2 s_{xx}^2 s_{yy}^2 \tag{2.14}$$

$$J_i = \begin{pmatrix} 2(x_i - b_x) s_{yy}^2 s_{zz}^2 \\ 2(y_i - b_y) s_{xx}^2 s_{zz}^2 \\ 2(z_i - b_z) s_{yy}^2 s_{xx}^2 \\ 2 s_{xx} s_{yy}^2 s_{zz}^2 - 2(y_i - b_y)^2 s_{xx} s_{zz}^2 - 2(z_i - b_z)^2 s_{xx} s_{yy}^2 \\ 2 s_{xx}^2 s_{yy} s_{zz}^2 - 2(x_i - b_x)^2 s_{yy} s_{zz}^2 - 2(z_i - b_z)^2 s_{yy} s_{xx}^2 \\ 2 s_{xx}^2 s_{yy}^2 s_{zz} - 2(x_i - b_x)^2 s_{zz} s_{yy}^2 - 2(y_i - b_y)^2 s_{zz} s_{xx}^2 \end{pmatrix}^T \tag{2.15}$$

Utilizing these equations, the Gauss-Newton algorithm is marched forward from a starting estimate until convergence. The starting estimate was initially set to the value expected from a perfect sensor, but this had a tendency to get stuck in a local minimum and drive the scale to 0. Starting the bias at 0 and the scale at twice the expected value resolved this.

Using the above methods to calibrate the three sensors, the data is adjusted and shown below in figures 2.1, 2.2, and 2.3. For brevity, unless needed for clarity, all sensor outputs will be assumed to have been calibrated via the preceding methods, and will not be indicated with a “cal” subscript.

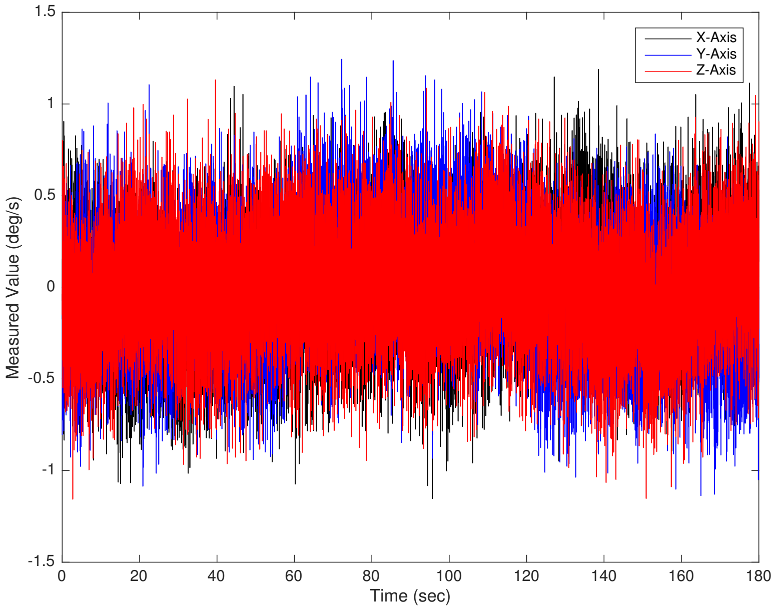

Figure 2.1: Calibrated gyroscope output in 6 sensor orientations

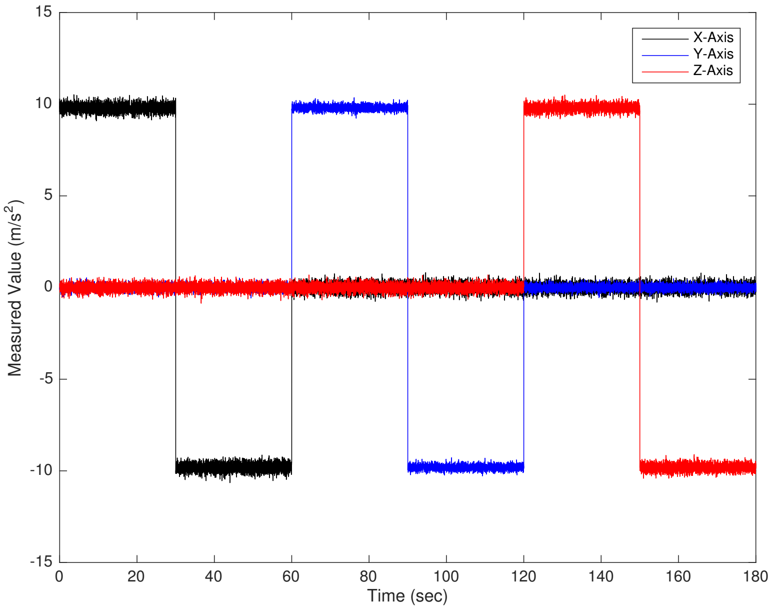

Figure 2.2: Calibrated accelerometer output in 6 sensor orientations

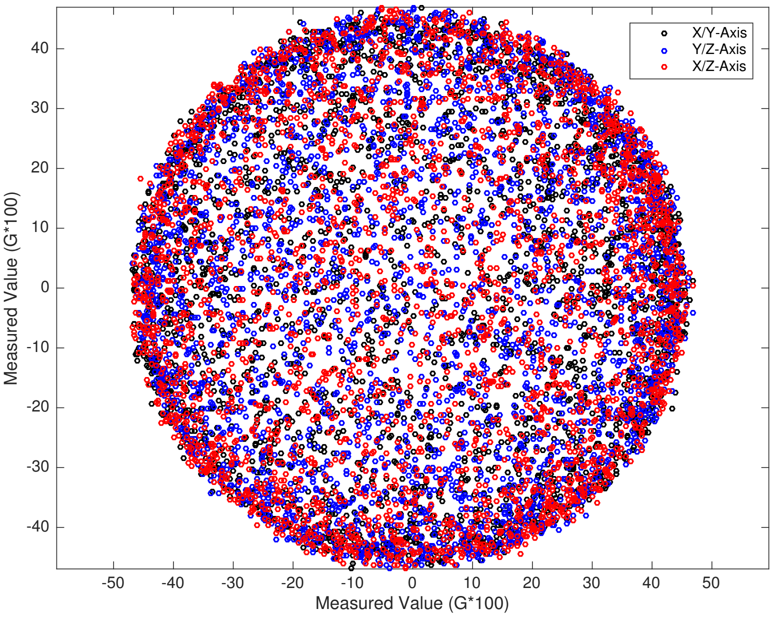

Figure 2.3: Calibrated magnetometer output rotated randomly

Chapter 3

Quaternion Algebra

A quaternion is an extension of complex numbers. Just as a complex number has a real part and an imaginary part $\boldsymbol{i}$, a quaternion has a real part and three imaginary parts: $\boldsymbol{i}$, $\boldsymbol{j}$, and $\boldsymbol{k}$. Notations vary between texts; this thesis will follow the JPL proposed standard convention [7], using the first three values as the imaginary part and the last value as the real part. Quaternions will be notated as a letter with an overbar, ie $\bar{q}$[1]:

$$\bar{q} = \begin{pmatrix} q_1\boldsymbol{i} \\ q_2\boldsymbol{j} \\ q_3\boldsymbol{k} \\ q_4 \end{pmatrix} \tag{3.1}$$

Much like complex numbers, the square of any of the imaginary parts of a quaternion is equal to -1. Multiplying two different imaginary parts is noncommutative; reversing order changes the sign. The imaginary combinations are:

$$\begin{aligned} \boldsymbol{i}^2 &= -1 & \boldsymbol{ij} &= -\boldsymbol{k} & \boldsymbol{ji} &= \boldsymbol{k} \\ \boldsymbol{j}^2 &= -1 & \boldsymbol{jk} &= -\boldsymbol{i} & \boldsymbol{kj} &= \boldsymbol{i} \\ \boldsymbol{k}^2 &= -1 & \boldsymbol{ki} &= -\boldsymbol{j} & \boldsymbol{ik} &= \boldsymbol{j} \end{aligned} \tag{3.2}$$

[1] The quaternion identities in this section follow from [8]

Note that this definition differs from that which Hamilton used.

As the multiplication of imaginary parts of a quaternion is noncommutative, quaternion multiplication is also noncommutative. The product of two quaternions can be derived by multiplying each element of two quaternions:

$$\begin{aligned}\bar{q} \otimes \bar{p} = {} & q_1 p_1 \boldsymbol{i}^2 + q_1 p_2 \boldsymbol{i}\boldsymbol{j} + q_1 p_3 \boldsymbol{i}\boldsymbol{k} + q_1 p_4 \boldsymbol{i} \\ & + q_2 p_1 \boldsymbol{j}\boldsymbol{i} + q_2 p_2 \boldsymbol{j}^2 + q_2 p_3 \boldsymbol{j}\boldsymbol{k} + q_2 p_4 \boldsymbol{j} \\ & + q_3 p_1 \boldsymbol{k}\boldsymbol{i} + q_3 p_2 \boldsymbol{k}\boldsymbol{j} + q_3 p_3 \boldsymbol{k}^2 + q_3 p_4 \boldsymbol{k} \\ & + q_4 p_1 \boldsymbol{i} + q_4 p_2 \boldsymbol{j} + q_4 p_3 \boldsymbol{k} + q_4 p_4\end{aligned} \tag{3.3}$$

Substituting the definitions of individual complex products (3.2):

$$\begin{aligned}\bar{q} \otimes \bar{p} = {} & - q_1 p_1 - q_1 p_2 \boldsymbol{k} + q_1 p_3 \boldsymbol{j} + q_1 p_4 \boldsymbol{i} \\ & + q_2 p_1 \boldsymbol{k} - q_2 p_2 - q_2 p_3 \boldsymbol{i} + q_2 p_4 \boldsymbol{j} \\ & - q_3 p_1 \boldsymbol{j} + q_3 p_2 \boldsymbol{i} - q_3 p_3 + q_3 p_4 \boldsymbol{k} \\ & + q_4 p_1 \boldsymbol{i} + q_4 p_2 \boldsymbol{j} + q_4 p_3 \boldsymbol{k} + q_4 p_4\end{aligned} \tag{3.4}$$

Then grouping the individual complex components into a new quaternion:

$$\bar{q} \otimes \bar{p} = \begin{pmatrix} (+q_1 p_4 - q_2 p_3 + q_3 p_2 + q_4 p_1)\boldsymbol{i} \\ (+q_1 p_3 + q_2 p_4 - q_3 p_1 + q_4 p_2)\boldsymbol{j} \\ (-q_1 p_2 + q_2 p_1 + q_3 p_4 + q_4 p_3)\boldsymbol{k} \\ -q_1 p_1 - q_2 p_2 - q_3 p_3 + q_4 p_4 \end{pmatrix} \tag{3.5}$$

The multiplication of the two quaternions is therefore equivalent to:

$$\bar{q} \otimes \bar{p} = \begin{bmatrix} q_4 & q_3 & -q_2 & q_1 \\ -q_3 & q_4 & q_1 & q_2 \\ q_2 & -q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} \equiv \mathcal{L}(\bar{q})\bar{p} \tag{3.6}$$

$$\mathcal{L}(\bar{q}) \equiv \begin{bmatrix} \Psi(\bar{q}) & \bar{q} \end{bmatrix} \tag{3.7}$$

$$\bar{q} \otimes \bar{p} = \begin{bmatrix} p_4 & -p_3 & p_2 & p_1 \\ p_3 & p_4 & -p_1 & p_2 \\ -p_2 & p_1 & p_4 & p_3 \\ -p_1 & -p_2 & -p_3 & p_4 \end{bmatrix} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} \equiv \mathcal{R}(\bar{p})\bar{q} \tag{3.8}$$

$$\mathcal{R}(\bar{p}) \equiv \begin{bmatrix} \Xi(\bar{p}) & \bar{p} \end{bmatrix} \tag{3.9}$$

$$\mathcal{L}(\bar{q})\bar{p} \equiv \mathcal{R}(\bar{p})\bar{q} \tag{3.10}$$

The Ξ matrix has a few useful properties that will be leveraged in chapter 5, namely:

$$\Xi(\bar{q})^T \Xi(\bar{q}) = I_{3\times 3} \tag{3.11}$$

$$\Xi(\bar{q})\Xi(\bar{q})^T = I_{4\times 4} - \bar{q}\bar{q}^T \tag{3.12}$$

$$\Xi(\bar{q})^T \bar{q} = 0 \tag{3.13}$$

Multiple products are as follows:

$$\begin{aligned} \bar{p} \otimes \bar{q} \otimes \bar{r} =& \mathcal{L}(\bar{p})\mathcal{L}(\bar{q})\bar{r} \\ =& \mathcal{L}(\bar{p})\mathcal{R}(\bar{r})\bar{q} \\ =& \mathcal{R}(\bar{r})\mathcal{L}(\bar{p})\bar{q} \\ =& \mathcal{R}(\bar{r})\mathcal{R}(\bar{q})\bar{p} \end{aligned} \tag{3.14}$$

The dot product between two quaternions is calculated the same way as a vector dot product.

$$\bar{q} \cdot \bar{p} = \bar{q}^T \bar{p} = q_1 p_1 + q_2 p_2 + q_3 p_3 + q_4 p_4 \tag{3.15}$$

The identity quaternion $\bar{I}$ has real part equal to 1, and all imaginary parts equal to 0.

$$\bar{I} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{3.16}$$

A pure quaternion has real part 0, and any value for the imaginary parts. A pure quaternion will generally be built from a vector; the pure quaternion made from a vector $\mathbf{v}$ will be:

$$\bar{\mathbf{v}} = \begin{bmatrix} \mathbf{v} \\ 0 \end{bmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ 0 \end{pmatrix} \tag{3.17}$$

The quaternion conjugate is found by multiplying all imaginary parts by -1:

$$\bar{q}^* = \begin{pmatrix} -q_1 \\ -q_2 \\ -q_3 \\ q_4 \end{pmatrix} \tag{3.18}$$

The norm of a quaternion is the standard vector euclidian norm:

$$\|\bar{q}\| = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} \tag{3.19}$$

The inverse of a quaternion is the conjugate divided by the norm:

$$\bar{q}^{-1} = \frac{\bar{q}^*}{\|\bar{q}\|} \tag{3.20}$$

Unit quaternions are defined as having a norm of 1, and hence their inverse is just their conjugate:

$$\|\bar{q}\| = 1 \Leftrightarrow \bar{q}^{-1} = \bar{q}^* \tag{3.21}$$

Unit quaternions can be used to define three dimensional rotations, much like a rotation matrix. According to Euler's rotation theorem, any rotation in three dimensional space can be defined by a single rotation about some fixed axis. Defining that axis as $\boldsymbol{\beta}$, a quaternion rotation can be defined as a rotation of angle α about $\boldsymbol{\beta}$:

$$\bar{q}_{rotation} = \begin{pmatrix} \sin\frac{\alpha}{2}\cos\beta_x \\ \sin\frac{\alpha}{2}\cos\beta_y \\ \sin\frac{\alpha}{2}\cos\beta_z \\ \cos\frac{\alpha}{2} \end{pmatrix} \tag{3.22}$$

Note that since a positive rotation about a positive axis is equivalent to a negative rotation about a negative axis, quaternions are not unique to a given rotation. Each quaternion has a single corresponding quaternion that represents the same rotation, that is $\bar{q} \equiv -\bar{q}$. Just as with rotation matrices, multiplying a quaternion from the left by a second quaternion rotates the first quaternion by the second. Hence in the same manner that a rotation matrix can be calculated via the multiplication of 3 rotation matrices from right (first rotation) to left (last rotation) of yaw, pitch, and roll, so too can a quaternion be calculated:

$$\bar{q}_\phi = \begin{pmatrix} \sin\frac{\phi}{2} \\ 0 \\ 0 \\ \cos\frac{\phi}{2} \end{pmatrix} \bar{q}_\theta = \begin{pmatrix} 0 \\ \sin\frac{\theta}{2} \\ 0 \\ \cos\frac{\theta}{2} \end{pmatrix} \bar{q}_\psi = \begin{pmatrix} 0 \\ 0 \\ \sin\frac{\psi}{2} \\ \cos\frac{\psi}{2} \end{pmatrix} \tag{3.23}$$

$$\bar{q}_{n\to b} = \bar{q}_\phi \otimes \bar{q}_\theta \otimes \bar{q}_\psi \tag{3.24}$$

For small angles, this may be approximated with:

$$\bar{q}_{n\to b} \approx \begin{pmatrix} \frac{\phi}{2} \\ \frac{\theta}{2} \\ \frac{\psi}{2} \\ 1 \end{pmatrix} \tag{3.25}$$

Note that this approximation results in a non unit-length quaternion.

Vectors can be rotated from the local frame to the body frame by a quaternion by forming a pure quaternion with the vector of interest as the imaginary part:

$$\bar{\mathbf{v}}^n = \begin{pmatrix} v_x \\ v_y \\ v_z \\ 0 \end{pmatrix} \tag{3.26}$$

The pure quaternion is multipied by the rotation quaternion on the left and the rotation quaternion inverse (quaternion conjugate for unit quaternion) on the right. This will result in another pure quaternion, and the rotated vector may be extracted from the imaginary part of the resultant quaternion.

$$\bar{\mathbf{v}}^b = \bar{q}_{n\to b}\bar{\mathbf{v}}^n\bar{q}^*_{n\to b} \tag{3.27}$$

Using this definition, the corresponding rotation matrix can easily be found.

$$\begin{pmatrix}\mathbf{v}^b\\0\end{pmatrix} = \mathcal{L}(\bar{q})\mathcal{R}(\bar{q}^*)\begin{pmatrix}\mathbf{v}^n\\0\end{pmatrix} \tag{3.28}$$

$$= \mathcal{R}(\bar{q}^*)\mathcal{L}(\bar{q})\begin{pmatrix}\mathbf{v}^n\\0\end{pmatrix} \tag{3.29}$$

$$= \begin{bmatrix}\Xi(\bar{q}^*) & \bar{q}^*\end{bmatrix}\begin{bmatrix}\Psi(\bar{q}) & \bar{q}\end{bmatrix}\begin{pmatrix}\mathbf{v}^n\\0\end{pmatrix} \tag{3.30}$$

$$= \begin{bmatrix}\Xi(\bar{q}^*) & \bar{q}^*\end{bmatrix}\begin{bmatrix}q_4 & q_3 & -q_2\\ -q_3 & q_4 & q_1\\ q_2 & -q_1 & q_4\\ -q_1 & -q_2 & -q_3\end{bmatrix}\mathbf{v}^n \tag{3.31}$$

$$= \begin{bmatrix}q_4 & q_3 & -q_2 & -q_1\\ -q_3 & q_4 & q_1 & -q_2\\ q_2 & -q_1 & q_4 & -q_3\\ q_1 & q_2 & q_3 & q_4\end{bmatrix}\begin{bmatrix}q_4 & q_3 & -q_2\\ -q_3 & q_4 & q_1\\ q_2 & -q_1 & q_4\\ -q_1 & -q_2 & -q_3\end{bmatrix}\mathbf{v}^n \tag{3.32}$$

$$\begin{pmatrix}\mathbf{v}^b\\0\end{pmatrix} = \begin{bmatrix}* & * & *\\ * & * & *\\ * & * & *\\ 0 & 0 & 0\end{bmatrix}\mathbf{v}^n \tag{3.33}$$

$$\mathbf{v}^b = \begin{bmatrix}q_4 & q_3 & -q_2 & -q_1\\ -q_3 & q_4 & q_1 & -q_2\\ q_2 & -q_1 & q_4 & -q_3\end{bmatrix}\begin{bmatrix}q_4 & q_3 & -q_2\\ -q_3 & q_4 & q_1\\ q_2 & -q_1 & q_4\\ -q_1 & -q_2 & -q_3\end{bmatrix}\mathbf{v}^n \tag{3.34}$$

$$\mathbf{v}^b = \Xi(\bar{q})^T\Psi(\bar{q})\mathbf{v}^n \tag{3.35}$$

$$C_{n\to b}(\bar{q}_{n\to b}) = \Xi(\bar{q}_{n\to b})^T\Psi(\bar{q}_{n\to b}) \tag{3.36}$$

The rotation quaternion can be found from the rotation matrix through one of the following relationships, where T is the trace of the rotation matrix.

$$\bar{q}_{n\to b}(C_{n\to b}) = \begin{pmatrix} \sqrt{(1+2C_{11}-T)/4} \\ (C_{12}+C_{21})/(4q_1) \\ (C_{13}+C_{31})/(4q_1) \\ (C_{23}-C_{32})/(4q_1) \end{pmatrix} = \begin{pmatrix} (C_{12}+C_{21})/(4q_2) \\ \sqrt{(1+2C_{22}-T)/4} \\ (C_{23}+C_{32})/(4q_2) \\ (C_{31}-C_{13})/(4q_2) \end{pmatrix} \tag{3.37}$$

$$= \begin{pmatrix} (C_{13}+C_{31})/(4q_3) \\ (C_{23}+C_{32})/(4q_3) \\ \sqrt{(1+2C_{33}-T)/4} \\ (C_{12}-C_{21})/(4q_3) \end{pmatrix} = \begin{pmatrix} (C_{23}+C_{32})/(4q_4) \\ (C_{31}+C_{13})/(4q_4) \\ (C_{12}-C_{21})/(4q_4) \\ \sqrt{(1+T)/4} \end{pmatrix} \tag{3.38}$$

Using a pivotal element of zero will result in a singular quaternion, but at least one of the pivotal elements will be nonzero. For best numerical accuracy, the form with the largest pivotal element should be used. This corresponds to the largest value of C_{11}, C_{22}, C_{33}, and T.

Chapter 4

Attitude Estimation with Complementary Filter

Using the calibrated gyroscope, accelerometer, and magnetometer output, the objective is to find an estimate of the attitude of the vehicle, which can be represented by a rotation quaternion $\bar{q}_{n \rightarrow b}$. The most straightforward way to obtain an attitude would be to integrate the output from the gyroscopes. As this estimate integrates all errors, the solution diverges quickly and is therefore useless on its own. A solution that does not diverge can be found by comparing the measured gravity vector and magnetic field vector to the assumed gravity and magnetic field vector, then solving for the rotation to align the two. The measured body-frame magnetic field vector is simply the calibrated output from the magnetometer. The measured body-frame gravity vector is solved for via the accelerometer, gyroscope, and airspeed measurements. From (1.7) and (1.9), the measurement of the gravity vector is obtained by:

$$\tilde{\mathbf{g}}^b = \tilde{\mathbf{a}}^b - \tilde{\mathbf{f}}^b \tag{4.1}$$

$$\mathbf{a}^b = \dot{\mathbf{v}}^b + \Omega \mathbf{v}^b \tag{4.2}$$

$$\dot{\mathbf{v}}^b \approx 0 \tag{4.3}$$

$$\mathbf{v}^b \approx \begin{pmatrix} \tilde{v} \\ 0 \\ 0 \end{pmatrix} \tag{4.4}$$

$$\tilde{\mathbf{a}}^b = \begin{pmatrix} 0 \\ \tilde{\omega}_z \tilde{v} \\ -\tilde{\omega}_y \tilde{v} \end{pmatrix} \tag{4.5}$$

$$\tilde{\mathbf{g}}^b = \begin{pmatrix} -\tilde{a}^x \\ \tilde{\omega}_z \tilde{v} - \tilde{a}^y \\ -\tilde{\omega}_y \tilde{v} - \tilde{a}^z \end{pmatrix} \tag{4.6}$$

Utilizing the magnetic field vector, the gravity vector, and the cross product of the two, an estimate of the rotation matrix can be found [4, p. 340]:

$$\tilde{C}_{n\to b} \begin{bmatrix} | & | & | \\ \hat{\mathbf{g}}^n & \hat{\mathbf{m}}^n & \hat{\mathbf{g}}^n \times \hat{\mathbf{m}}^n \\ | & | & | \end{bmatrix} = \begin{bmatrix} | & | & | \\ \tilde{\mathbf{g}}^b & \tilde{\mathbf{m}}^b & \tilde{\mathbf{g}}^b \times \tilde{\mathbf{m}}^b \\ | & | & | \end{bmatrix} \tag{4.7}$$

$$\tilde{C}_{n\to b} = \begin{bmatrix} | & | & | \\ \tilde{\mathbf{g}}^b & \tilde{\mathbf{m}}^b & \tilde{\mathbf{g}}^b \times \tilde{\mathbf{m}}^b \\ | & | & | \end{bmatrix} \begin{bmatrix} | & | & | \\ \hat{\mathbf{g}}^n & \hat{\mathbf{m}}^n & \hat{\mathbf{g}}^n \times \hat{\mathbf{m}}^n \\ | & | & | \end{bmatrix}^{-1} \tag{4.8}$$

As this calculation depends on the matrix of gravity and magnetic field vectors to be nonsingular, the determinant is:

$$\left| \begin{bmatrix} | & | & | \\ \hat{\mathbf{g}}^n & \hat{\mathbf{m}}^n & \hat{\mathbf{g}}^n \times \hat{\mathbf{m}}^n \\ | & | & | \end{bmatrix} \right| = \left| \begin{bmatrix} 0 & m_x & -gm_y \\ 0 & m_y & gm_x \\ g & m_z & 0 \end{bmatrix} \right| = g^2(m_x^2 + m_y^2) \tag{4.9}$$

Hence the matrix is singular if and only if the local magnetic field vector points straight up or down. This is as expected, as this would cause the magnetic field and gravity vectors to be coincident, and therefore linearly dependent.

The resulting $\tilde{C}_{n\to b}$ is unlikely to be orthonormal; in order to find the closest orthonormal matrix to the computed matrix, a Singular Value Decomposition (SVD) is computed. The SVD decomposes the rotation matrix into UΣV*, where U and V* are orthonormal matrices that are effectively rotation matrices, and Σ is a scaling factor. By setting $\Sigma = I$, the closest orthonormal rotation matrix is computed as UV*.

While the attitude estimate based on aligning the gravity and magnetic field vectors won't diverge over time, it is subject to a significant amount of high frequency noise due to the AWGN on all of the sensors; other factors not modeled, such as turbulence, would also contribute to the noise on this measurement. Luckily, gyroscopic integration is much more stable over short-term estimates. By fusing the data together, a much more robust estimation of the air vehicle attitude can be found than either estimate gives individually.

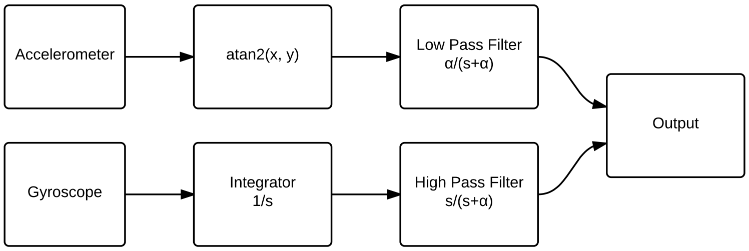

Figure 4.1: Complementary filter

A common method to fuse gyroscopes and accelerometers in two dimensions is a complementary filter. The complementary filter combines a high-pass filter on the integrated gyroscope output, and a low-pass filter on the angle found from the accelerometer output as shown in figure 4.1. Adding the transfer functions of the two must equal 1, indicating the total gain is unity. The transfer functions for the high-pass and low-pass filters are:

$$\mathrm{TF}_{\mathrm{HP}} = \frac{s}{s+\alpha} \tag{4.10}$$

$$\mathrm{TF}_{\mathrm{LP}} = 1 - \frac{s}{s+\alpha} = \frac{\alpha}{s+\alpha} \tag{4.11}$$

The angles measured by using a first order numerical integration of the gyroscope and finding the acceleration vector from the accelerometer are:

$$\tilde{\theta}^{k}_{gyro} = \tilde{\theta}^{k-1}_{gyro} + \Delta_t \tilde{\dot{\theta}} \tag{4.12}$$

$$\tilde{\theta}_{accel} = \mathrm{atan2}(\tilde{a}_1, \tilde{a}_2) \tag{4.13}$$

The calculated integrated gyroscope angle and acceleration vector angle are then inserted into their respective high-pass and low-pass filters:

$$\hat{\theta}^{k} = \alpha\hat{\theta}^{k-1} + \alpha(\overbrace{\tilde{\theta}^{k}_{gyro} - \tilde{\theta}^{k-1}_{gyro}}^{\Delta_t \tilde{\dot{\theta}}}) \tag{4.14}$$

$$\hat{\theta}^{k} = \alpha\hat{\theta}^{k-1} + (1-\alpha)\tilde{\theta}_{accel} \tag{4.15}$$

The high-pass filter uses the first order derivative of the integrated gyroscope output, which by rearranging the terms in (4.12) can be seen to be the gyroscope output multiplied by the timestep. Using linearity, the two filters are then combined into the complementary filter:

$$\hat{\theta}^{k} = \alpha\,\underbrace{(\hat{\theta}^{k-1} + \Delta_t \tilde{\dot{\theta}})}_{1} + (1-\alpha)\,\underbrace{\tilde{\theta}_{accel}}_{2} \tag{4.16}$$

This type of filter is very simple to implement, yet effective at finding a reasonable estimate of the angle. It is desired to use a similar filter design for a full attitude solution. The filter effectively interpolates between the previous estimate updated with gyroscopic measurements in part 1 and the estimate based on the accelerometer in part 2, with a higher *k* factor giving more weight to the gyroscopic update. While interpolation can not be easily computed between two rotation matrices, it can be easily computed between two quaternions.

The quaternion equivalent to (4.12) is:

$$\tilde{\bar{q}}^{k}_{n\to b} = \tilde{\bar{q}}_{gyro} \otimes \tilde{\bar{q}}^{k-1}_{n\to b} \tag{4.17}$$

$\tilde{\bar{q}}_{n\to b}$ represents the integrated value, and $\tilde{\bar{q}}_{gyro}$ is the quaternion that rotates the previous integrated value by the new integrated value; $\tilde{\bar{q}}_{gyro}$ can be found simply using the small angle approximation and first order integrator:

$$\tilde{\bar{q}}_{gyro} = \begin{pmatrix} \frac{1}{2}\Delta_t\tilde{\omega}_x \\ \frac{1}{2}\Delta_t\tilde{\omega}_y \\ \frac{1}{2}\Delta_t\tilde{\omega}_z \\ 1 \end{pmatrix} \tag{4.18}$$

Likewise, the analog to (4.13) is:

$$\tilde{\bar{q}}_{accel} = \bar{q}_{n\to b}(\tilde{C}_{n\to b}) \tag{4.19}$$

This converts the measured rotation matrix $\tilde{C}_{n\to b}$ calculated in (4.7) to a quaternion.

Using the final form of the complementary filter in (4.16), the analogous Spherical Linear intERPolation (Slerp) is applied to interpolate between the integration quaternion $\tilde{\bar{q}}_{gyro}$ and $\tilde{\bar{q}}_{accel}$ via (4.20), where $\Omega = \cos^{-1}(\bar{q}_1 \cdot \bar{q}_2)$ is the angle subtended by the arc, and α is the interpolation parameter. It should be noted that since $\bar{q} = -\bar{q}$ in representation of rotation, the angles are checked to see which of $\pm\bar{q}_2$ is closer to $\bar{q}_1$, to avoid interpolating the long way around. Additionally, the trivial case where $\bar{q}_1 = \pm\bar{q}_2$ is undefined by the Slerp expression; since they are equivalent, no evaluation is necessary. [3]

$$\bar{q}_{slerp} = \frac{\sin[\alpha\Omega]}{\sin\Omega}\bar{q}_1 + \frac{\sin[(1-\alpha)\Omega]}{\sin\Omega}\bar{q}_2 \tag{4.20}$$

$$\hat{\bar{q}}^{k}_{n\to b} = \frac{\sin[\alpha\Omega]}{\sin\Omega}\tilde{\bar{q}}_{gyro} \otimes \hat{\bar{q}}^{k-1}_{n\to b} + \frac{\sin[(1-\alpha)\Omega]}{\sin\Omega}\tilde{\bar{q}}_{accel} \tag{4.21}$$

There are now three identified methods for tracking attitude: Gyroscope Integration, Magnetic/Gravity Vector alignment, and a complementary filter using the first two measurements. Figures 4.3, 4.2, and 4.4 show the results of all three approximations with the plant through a series of maneuvers. First a standard rate (3 deg/sec) coordinated turn is made to the right, covering 360 degrees. After that, a steady heading sideslip is made at 24 degrees of bank. Then a loop is completed, followed by a roll.

The integrated value diverges as expected, and is clearly of little use on its own. The estimate obtained from the gravity and magnetic field vector follows the plant throughout the entire simulation, but it contains a significant amount of high-frequency noise. The complementary filter does a good job of fusing the two measurements together to get a reasonable estimate of the state. The biggest shortcoming of this approach is the lack of an estimate of the gyroscope bias; as time goes on and the bias drifts, the filter will lose accuracy. The extended Kalman filter overcomes this, and is explored in chapter 5.

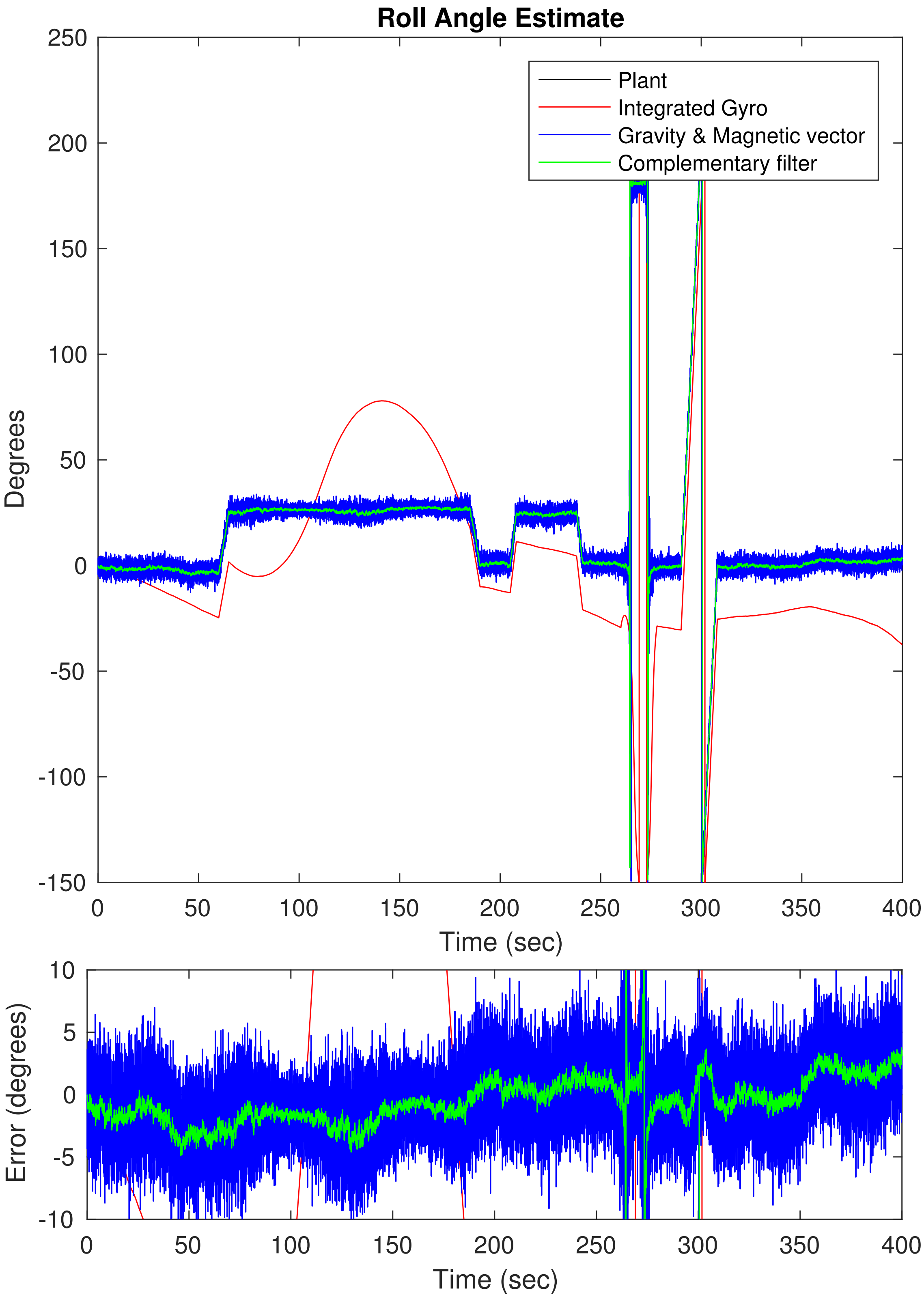

Figure 4.2: Roll estimate through several maneuvers

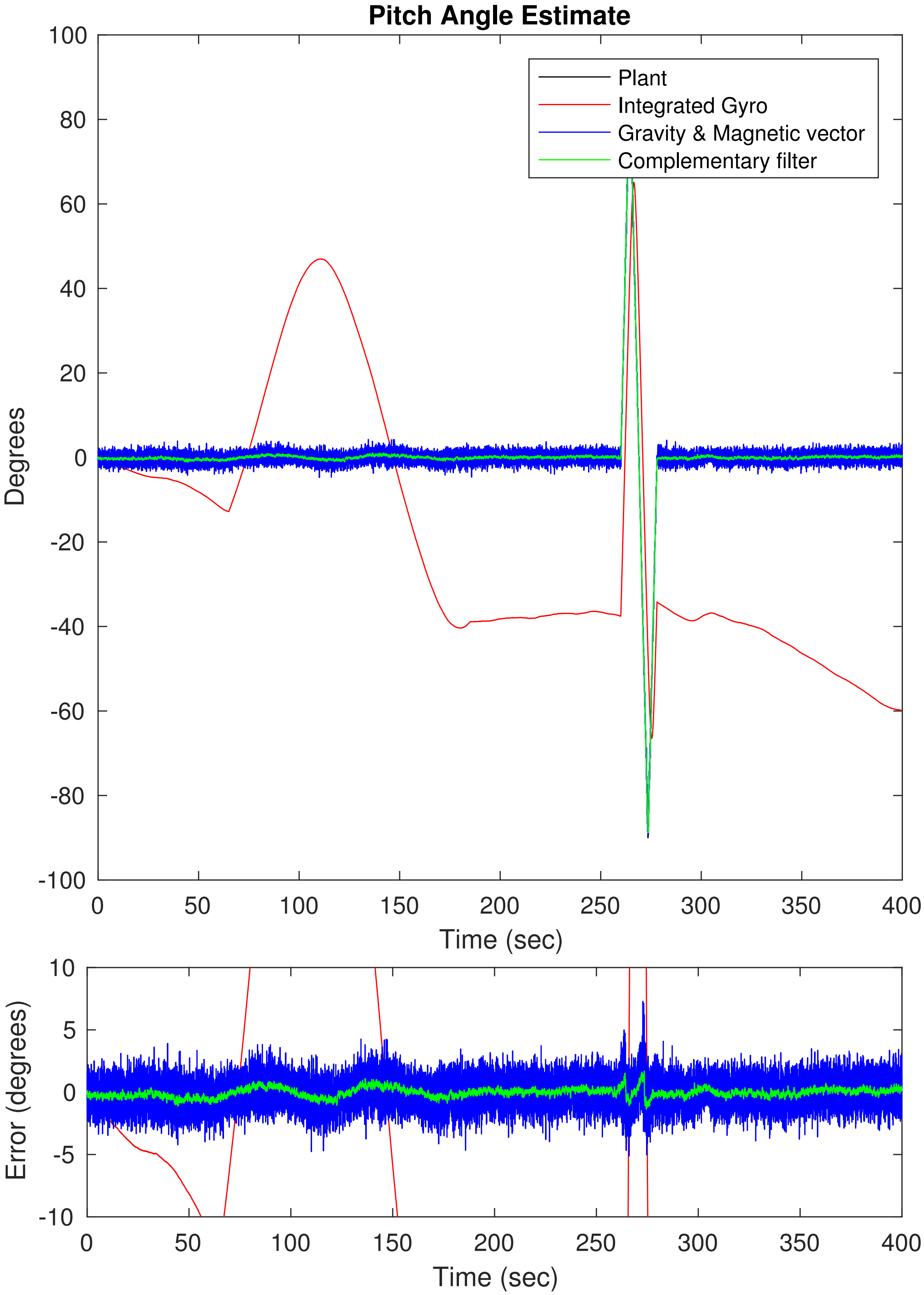

Figure 4.3: Pitch estimate through several maneuvers

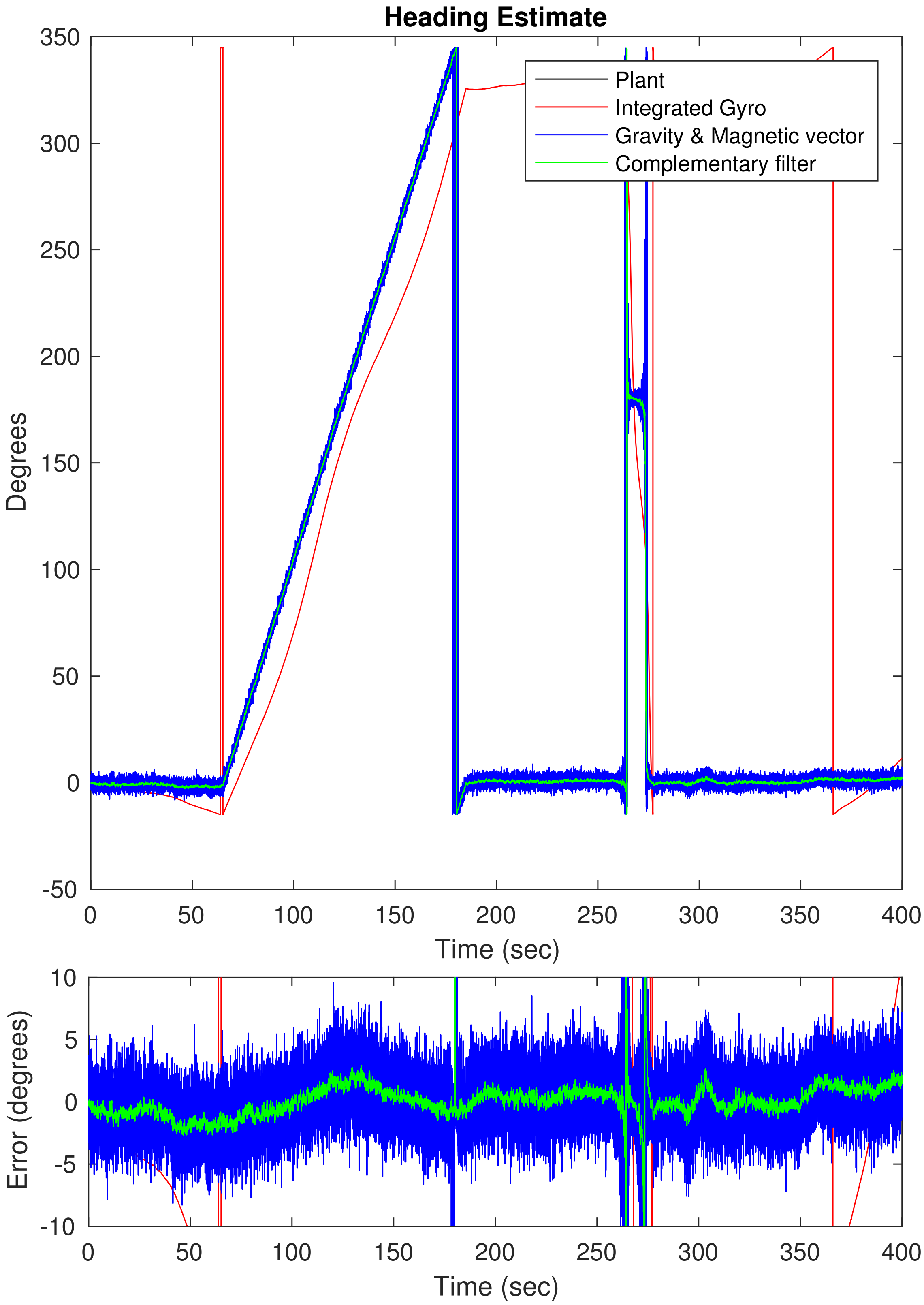

Figure 4.4: Heading estimate through several maneuvers

Chapter 5

Extended Kalman Filter (EKF)

While the complementary filter does reasonably well with attitude estimation, it does not estimate errors; the only error correction comes from the calibration routine, and this not only fails to capture all of the errors, it also becomes less relevant over time as the gyroscope drifts. Using an extended Kalman filter will allow estimation of the gyroscope bias over time, leading to a stable solution that does not drift with gyroscope bias.

The Kalman filter can be derived as follows[1]. Assuming a state space discrete-time linear time-varying (LTV) system with no inputs, and uncorrelated zero-mean AWGN $\mathbf{w}_k$ and $\mathbf{v}_k$ of the form:

$$\mathbf{x}_k = F_k \mathbf{x}_{k-1} + G_k \mathbf{w}_k \tag{5.1}$$

$$\mathbf{y}_k = H_k \mathbf{x}_k + \mathbf{v}_k \tag{5.2}$$

The expected value $\mathbf{x}$ of the state after one time step is simply:

$$E(\hat{\mathbf{x}}_k) = F_k \hat{\mathbf{x}}_{k-1} \tag{5.3}$$

[1] The derivation of the Kalman filter follows [9].

Defining the state error $\Delta\mathbf{x}$:

$$\Delta\mathbf{x} = \mathbf{x} - \hat{\mathbf{x}} \tag{5.4}$$

The covariance matrix P is the covariance of the state, or the expected value of the outer product of the error:

$$P_k = E(\Delta\mathbf{x}_k \Delta\mathbf{x}_k^T) \tag{5.5}$$

As the system is linear, the expected value of the error and covariance after one time step is:

$$\Delta\mathbf{x}_k = F_k(\Delta\mathbf{x}_{k-1}) + G_k\mathbf{w}_k \tag{5.6}$$

$$P_k = E[(F_k(\Delta\mathbf{x}_{k-1}) + G_k\mathbf{w}_k)(F_k(\Delta\mathbf{x}_{k-1}) + G_k\mathbf{w}_k)^T] \tag{5.7}$$

$$\begin{aligned} P_k = E[F_k &\overbrace{(\Delta\mathbf{x}_{k-1})(\Delta\mathbf{x}_{k-1})^T}^{P_{k-1}} F_k^T \\ &+ F_k(\Delta\mathbf{x}_{k-1})\mathbf{w}_k^T G_k^T \\ &+ G_k\mathbf{w}_k F_k(\Delta\mathbf{x}_{k-1})^T \\ &+ G_k \underbrace{\mathbf{w}_k\mathbf{w}_k^T}_{Q_k} G_k^T] \end{aligned} \tag{5.8}$$

As $\mathbf{w}_k$ is uncorrelated with $\Delta\mathbf{x}_{k-1}$, the second and third terms are 0, and the first and fourth terms simplify to:

$$P_k = F_k P_{k-1} F_k^T + G_k Q_k G_k^T \tag{5.9}$$

This can be propagated forward along with (5.3) to give the predict stages of the Kalman filter.

Measurements can be taken from any available sensors at any time; they do not need to be taken with every prediction stage. The measurements map to the state through the H matrix. The output from the sensor $\tilde{\mathbf{y}}_k$ can be compared to the estimated value from the state $\hat{\mathbf{y}}_k = H_k\hat{\mathbf{x}}_k$ to give the innovation $\mathbf{z}_k = \tilde{\mathbf{y}}_k - \hat{\mathbf{y}}_k$. This simplifies to:

$$\mathbf{z}_k = \tilde{\mathbf{y}}_k - H_k\hat{\mathbf{x}}_k \tag{5.10}$$

With the innovation, the covariance of the measurements R_k, and the state covariance P_k, the expected value of the state and state covariance can be updated. With $\hat{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ jointly Gaussian, $p_{\hat{\mathbf{x}}|\tilde{\mathbf{y}}} \sim \mathcal{N}(\mathbf{m}, \Sigma)$ with:

$$\mathbf{m} = \mathbf{m}_{\hat{\mathbf{x}}} + \Sigma_{\hat{\mathbf{x}}\tilde{\mathbf{y}}}\Sigma_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}}^{-1}(\tilde{\mathbf{y}} - \hat{\mathbf{y}}) \tag{5.11}$$

$$\Sigma = \Sigma_{\hat{\mathbf{x}}\hat{\mathbf{x}}} - \Sigma_{\hat{\mathbf{x}}\tilde{\mathbf{y}}}\Sigma_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}}^{-1}\Sigma_{\tilde{\mathbf{y}}\hat{\mathbf{x}}} \tag{5.12}$$

As $\tilde{\mathbf{y}}_k = H_k\mathbf{x}_k + \mathbf{v}_k$, $\begin{pmatrix}\mathbf{x}_k \\ \tilde{\mathbf{y}}_k\end{pmatrix}$ is Gaussian with mean and covariance:

$$\begin{pmatrix}\hat{\mathbf{x}}_k \\ H_k\hat{\mathbf{x}}_k\end{pmatrix}, \begin{bmatrix} P_k & P_kH_k^T \\ H_kP_k & H_kP_kH_k^T + R_k\end{bmatrix} \tag{5.13}$$

Utilizing superscripts on the state and covariance to indicate pre-($\hat{\mathbf{x}}^-$ and P^-) and post-($\hat{\mathbf{x}}^+$ and P^+) measurement update values, the measurements can be incorporated. Substituting the variables of (5.13) into (5.11) and (5.12), the optimal estimate for $\hat{\mathbf{x}}_k^+$ and P_k^+ can be found via:

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \overbrace{P_k^-H_k^T(H_kP_k^-H_k^T + R_k)^{-1}}^{L_k}\overbrace{(\tilde{\mathbf{y}}_k - \hat{\mathbf{y}}_k)}^{\mathbf{z}_k} \tag{5.14}$$

$$P_k^+ = P_k^- - \underbrace{P_k^-H_k^T(H_kP_k^-H_k^T + R_k)^{-1}}_{L_k}H_kP_k^- \tag{5.15}$$

These give the update equations of the Kalman filter.

Predict

$$\hat{\mathbf{x}}_k^- = F_k \hat{\mathbf{x}}_{k-1}^+ \tag{5.16}$$

$$P_k^- = F_k P_{k-1}^+ F_k^T + G_k Q_k G_k^T \tag{5.17}$$

Update

$$\mathbf{z}_k = \tilde{\mathbf{y}}_k - H_k \hat{\mathbf{x}}_k^- \tag{5.18}$$

$$L_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \tag{5.19}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + L_k \mathbf{z}_k \tag{5.20}$$

$$P_k^+ = (I - L_k H_k) P_k^- \tag{5.21}$$

The nonlinear variant, the extended Kalman filter (EKF), is found simply by replacing the predicted state estimate from (5.16) with the nonlinear state transition equation $f(\hat{\mathbf{x}}_{k-1}^+)$ and replacing the observation matrix from (5.10) with the nonlinear observation equation $h(\hat{\mathbf{x}}_k^-)$. The F and H matrices used in the covariance calculations are then the jacobian of f and h for the respective time step.

Predict

$$\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}^+) \tag{5.22}$$

$$F_k = \frac{\partial f}{\partial x}\Big|_{\hat{\mathbf{x}}_{k-1}^+} \tag{5.23}$$

$$P_k^- = F_k P_{k-1}^+ F_k^T + G_k Q_k G_k^T \tag{5.24}$$

Update

$$\hat{\mathbf{z}}_k = \tilde{\mathbf{y}}_k - h(\hat{\mathbf{x}}_k^-) \tag{5.25}$$

$$H_k = \frac{\partial h}{\partial x}\Big|_{\hat{\mathbf{x}}_k^-} \tag{5.26}$$

$$L_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \tag{5.27}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + L_k \mathbf{z}_k \tag{5.28}$$

$$P_k^+ = (I - L_k H_k) P_k^- \tag{5.29}$$

Defining the state of the EKF as the rotation quaternion $\bar{q}_{n\to b}$ and the gyroscope bias, and dropping the subscript from the quaternion[2]:

$$\mathbf{x}(t) = \begin{pmatrix} \bar{q}(t) \\ \mathbf{b}(t) \end{pmatrix} \tag{5.30}$$

Using the pure quaternion $\bar{\boldsymbol{\omega}}$ from the rotation rate vector $\boldsymbol{\omega}$, the rotation quaternion satisfies the following differential equation.

$$\frac{d}{dt}\bar{q}(t) = \frac{1}{2}\bar{\boldsymbol{\omega}}(t) \otimes \bar{q}(t) \tag{5.31}$$

Substituting the measured value $\boldsymbol{\omega} = \tilde{\boldsymbol{\omega}} - \mathbf{b}_\omega - \mathbf{w}_\omega$ gives the set of coupled differential equations for the state:

$$\frac{d}{dt}\bar{q}(t) = \frac{1}{2}\mathcal{L}(\bar{\tilde{\boldsymbol{\omega}}} - \bar{\mathbf{b}}_\omega - \bar{\mathbf{w}}_\omega)\bar{q}(t) \tag{5.32}$$

$$\frac{d}{dt}b(t) = \mathbf{w}_{\omega b} \tag{5.33}$$

Noting that from (3.7), (3.10), and (3.17), for any pure quaternion $\bar{\mathbf{v}}$:

$$\mathcal{L}(\bar{\mathbf{v}})\bar{q} = \mathcal{R}(\bar{q})\bar{\mathbf{v}} \tag{5.34}$$

$$= \begin{bmatrix} \Xi(\bar{q}) & \bar{q} \end{bmatrix} \begin{pmatrix} \mathbf{v} \\ 0 \end{pmatrix} \tag{5.35}$$

$$\mathcal{L}(\bar{\mathbf{v}})\bar{q} = \Xi(\bar{q})\mathbf{v} \tag{5.36}$$

[2] This implementation of the Quaternion EKF follows closely with [6], with expansion, clarification, and a different measurement approach.

The quaternion state equation can then be redefined as:

$$\frac{d}{dt}\bar{q}(t) = \frac{1}{2}\mathcal{L}(\tilde{\boldsymbol{\omega}})\bar{q}(t) - \frac{1}{2}\Xi(\bar{q}(t))\mathbf{b}_\omega(t) - \frac{1}{2}\Xi(\bar{q}(t))\mathbf{w}_\omega(t) \tag{5.37}$$

And hence the continuous-time state equation is:

$$\frac{d}{dt}\mathbf{x}(t) = F(t)\mathbf{x}(t) + G(t)\mathbf{w}(t) \tag{5.38}$$

$$F(t) = \begin{bmatrix} \frac{1}{2}\mathcal{L}(\tilde{\boldsymbol{\omega}}) & -\frac{1}{2}\Xi(\bar{q}) \\ 0_{3\times 4} & 0_{3\times 3} \end{bmatrix} \tag{5.39}$$

$$G(t) = \begin{bmatrix} -\frac{1}{2}\Xi(\bar{q}) & 0_{4\times 3} \\ 0_{3\times 3} & I_{3\times 3} \end{bmatrix} \tag{5.40}$$

$$\mathbf{w}(t) = \begin{pmatrix} \mathbf{w}_\omega(t) \\ \mathbf{w}_{\omega b}(t) \end{pmatrix} \tag{5.41}$$

Discretizing the equation with the assumption that the gyroscope output is constant over the integration interval, a quaternion is generated to rotate the state by the integrated angle. $\Xi(\bar{q})$ can easily be integrated, as the $\bar{q}$ is held constant at $\bar{q}_{k-1}$ over the timestep. With $\bar{q}(\hat{\boldsymbol{\omega}}_k)$ indicating the small angle approximation of the quaternion rotating by angle $\hat{\boldsymbol{\omega}}_k$ (3.25), holding $\bar{q}$ and $\tilde{\boldsymbol{\omega}}$ constant over the timestep at $\bar{q}_{k-1}$ and $\tilde{\boldsymbol{\omega}}_k$, and defining $\hat{\boldsymbol{\omega}}_k = \int_{t_{k-1}}^{t_k} \tilde{\boldsymbol{\omega}}\, dt$, the discretized equations are:

$$\bar{q}_k = \int_{t_0}^{t_k} \frac{1}{2}\mathcal{L}(\tilde{\boldsymbol{\omega}}(t))\bar{q}(t) - \frac{1}{2}\Xi(\bar{q}(t))\mathbf{b}(t)\, dt \tag{5.42}$$

$$= \underbrace{\int_{t_0}^{t_{k-1}} \frac{1}{2}\mathcal{L}(\tilde{\boldsymbol{\omega}}(t))\bar{q}(t) - \frac{1}{2}\Xi(\bar{q}(t))\mathbf{b}(t)\, dt}_{\bar{q}_{k-1}} + \int_{t_{k-1}}^{t_k} \frac{1}{2}\mathcal{L}(\tilde{\boldsymbol{\omega}}(t))\bar{q}(t) - \frac{1}{2}\Xi(\bar{q}(t))\mathbf{b}(t)\, dt \tag{5.43}$$

$$= \bar{q}_{k-1} + \frac{1}{2}\mathcal{L}(\hat{\boldsymbol{\omega}}_k)\bar{q}_{k-1} - \frac{1}{2}\Delta_t\Xi(\bar{q}_{k-1})\mathbf{b}_{k-1} \tag{5.44}$$

$$= (I + \frac{1}{2}\mathcal{L}(\hat{\boldsymbol{\omega}}_k))\bar{q}_{k-1} - \frac{1}{2}\Delta_t\Xi(\bar{q}_{k-1})\mathbf{b}_{k-1} \tag{5.45}$$

$$= \begin{bmatrix} 1 & \frac{1}{2}\hat{\omega}_3 & -\frac{1}{2}\hat{\omega}_2 & \frac{1}{2}\hat{\omega}_1 \\ -\frac{1}{2}\hat{\omega}_3 & 1 & \frac{1}{2}\hat{\omega}_1 & \frac{1}{2}\hat{\omega}_2 \\ \frac{1}{2}\hat{\omega}_2 & -\frac{1}{2}\hat{\omega}_1 & 1 & \frac{1}{2}\hat{\omega}_3 \\ -\frac{1}{2}\hat{\omega}_1 & -\frac{1}{2}\hat{\omega}_2 & -\frac{1}{2}\hat{\omega}_3 & 1 \end{bmatrix} \bar{q}_{k-1} - \frac{1}{2}\Delta_t \Xi(\bar{q}_{k-1})\mathbf{b}_{k-1} \tag{5.46}$$

$$= \mathcal{L}(\bar{q}(\hat{\boldsymbol{\omega}}_k))\bar{q}_{k-1} - \frac{1}{2}\Delta_t \Xi(\bar{q}_{k-1})\mathbf{b}_{k-1} \tag{5.47}$$

Giving the discrete time state transition matrix:

$$F_k = \begin{bmatrix} \mathcal{L}(\bar{q}(\hat{\boldsymbol{\omega}}_k)) & -\frac{1}{2}\Xi(\bar{q}_{k-1})\Delta_t \\ 0_{3\times4} & I_{3\times3} \end{bmatrix} \tag{5.48}$$

The small angle approximation of $\bar{q}(\hat{\boldsymbol{\omega}})$ leads to the introduction of several errors, including: a resulting quaternion that is no longer unit length; accuracy loss for angles larger than the small angle approximation of *sin* and *cos* are valid; and higher order errors from the assumption that the quaternion and rotation rates are held constant over the integration period. With small enough time steps for the rotation rates expected, the small angle and higher order errors become negligible. The approximation will, however, tend to increase the length of the quaternion over time, which will occasionally need to be divided by the quaternion norm in order to keep the unit length constraint.

Just as with the complementary filter, there are two vectors available for measurements: the vector from the magnetometer and the estimated gravity vector from the accelerometer, corrected with the gyroscope and velocity measurements (4.6). The nonlinear measurement function, based on the assumed-known local magnetic field vector, is:

$$\hat{\mathbf{y}}_k^m = h^m(\hat{\mathbf{x}}_k^-) = C_{n\to b}(\hat{\bar{q}}_k)\mathbf{m}^n \tag{5.49}$$

The jacobian for the magnetometer, H_k^m, is:

$$H_k^m = \frac{\partial h^m}{\partial x}\bigg|_{\hat{\mathbf{x}}_k^-} = 2\begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \mathbf{c}_3 & \mathbf{c}_4 & 0_{3\times3} \end{bmatrix} \tag{5.50}$$

$$\mathbf{c}_1 = \begin{pmatrix} m_xq_1 + m_yq_2 + m_zq_3 \\ m_xq_2 - m_yq_1 + m_zq_4 \\ m_xq_3 - m_yq_4 - m_zq_1 \end{pmatrix} \qquad \mathbf{c}_2 = \begin{pmatrix} -m_xq_2 + m_yq_1 - m_zq_4 \\ m_xq_1 + m_yq_2 + m_zq_3 \\ m_xq_4 + m_yq_3 - m_zq_2 \end{pmatrix}$$

$$\mathbf{c}_3 = \begin{pmatrix} -m_xq_3 + m_yq_4 + m_zq_1 \\ -m_xq_4 - m_yq_3 + m_zq_2 \\ m_xq_1 + m_yq_2 + m_zq_3 \end{pmatrix} \qquad \mathbf{c}_4 = \begin{pmatrix} m_xq_4 + m_yq_3 - m_zq_2 \\ -m_xq_3 + m_yq_4 + m_zq_1 \\ m_xq_2 - m_yq_1 + m_zq_4 \end{pmatrix}$$

The measurement matrix for the estimated gravity vector is similar to the magnetometer, but simpler due to the zero x and y components of the gravity vector:

$$\hat{\mathbf{y}}^a_k = h^a(\hat{\mathbf{x}}^-_k) = C_{n\to b}(\hat{\bar{q}}_k)\mathbf{g}^n \tag{5.51}$$

$$H^a_k = 2g \begin{bmatrix} q_3 & -q_4 & q_1 & -q_2 & 0 & 0 & 0 \\ q_4 & q_3 & q_2 & q_1 & 0 & 0 & 0 \\ -q_1 & -q_2 & q_3 & q_4 & 0 & 0 & 0 \end{bmatrix} \tag{5.52}$$

Assuming a small stochastic error $\Delta\bar{q}$ on the quaternion estimate $\hat{\bar{q}}_k$, due to the unit constraint on both the plant and estimate quaternion, the error will be nearly orthogonal to the deterministic estimate quaternion at every timestep, as shown in figure 5.1[3].

$$\Delta\bar{q}^T_k\hat{\bar{q}}_k \approx 0 \tag{5.53}$$

From that, it follows that $\begin{pmatrix} \hat{\bar{q}}_k \\ 0 \end{pmatrix}$ is a null vector of P:

$$\Delta x_k \Delta x^T_k = \begin{pmatrix} \Delta\bar{q}_k \\ \Delta\mathbf{b}_k \end{pmatrix} \begin{pmatrix} \Delta\bar{q}^T_k & \Delta\mathbf{b}^T_k \end{pmatrix} \tag{5.54}$$

$$\Delta x_k \Delta x^T_k \begin{pmatrix} \hat{\bar{q}}_k \\ 0 \end{pmatrix} = \begin{pmatrix} \Delta\hat{\bar{q}}_k \\ \Delta\mathbf{b}_k \end{pmatrix} \underbrace{\begin{pmatrix} \Delta\bar{q}^T_k & \Delta\mathbf{b}^T_k \end{pmatrix} \begin{pmatrix} \hat{\bar{q}}_k \\ 0 \end{pmatrix}}_{\approx 0} \tag{5.55}$$

$$E\left(\Delta x_k \Delta x^T_k \begin{pmatrix} \hat{\bar{q}}_k \\ 0 \end{pmatrix}\right) \approx 0 \tag{5.56}$$

[3] It it important to note that $\Delta\bar{q}$ is stochastic and $\hat{\bar{q}}_k$ is deterministic, but they will always be approximately perpendicular for small angles due to the unit constraint on both the quaternion plant and the quaternion estimate.

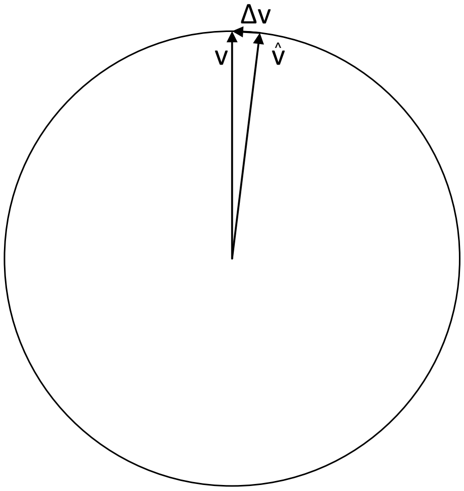

Figure 5.1: Unit constrained state, demonstrating that a small state error is nearly orthogonal to state estimate

$$P_k \begin{pmatrix} \hat{\bar{q}}_k \\ 0 \end{pmatrix} \approx 0 \tag{5.57}$$

The covariance matrix is therefore nearly singular. In the simulations conducted for this thesis, sensors with significant noise levels were used, and double precision floating point numbers were used in all calculations. The nearly singular covariance matrix has not been an issue due to both of these mitigating factors, although single or half precision floating point calculations and/or lower noise sensors will exacerbate the problem. In order to rectify these problems, a full-rank reduced representation of the covariance matrix is desired.

Rather than defining the quaternion error as the difference between the plant and the estimate, $\Delta\bar{q} = \bar{q} - \hat{\bar{q}}$, it can instead be defined as the quaternion required to rotate the estimate to the plant. Assuming this is a small angle, the information of interest is contained entirely within the vector part of the quaternion (the real part ≈ 1). Using (3.11) and (3.13), the error quaternion $\delta\bar{q}$ can be defined as:

$$\bar{q} = \delta\bar{q} \otimes \hat{\bar{q}} \tag{5.58}$$

$$= \mathcal{R}(\hat{\bar{q}})\delta\bar{q} \tag{5.59}$$

$$= \begin{bmatrix} \Xi(\hat{\bar{q}}) & \hat{\bar{q}} \end{bmatrix} \begin{pmatrix} \delta\mathbf{q} \\ \delta q_4 \end{pmatrix} \tag{5.60}$$

$$= \Xi(\hat{\bar{q}})\delta\mathbf{q} + \hat{\bar{q}}\delta q_4 \tag{5.61}$$

$$\Xi(\hat{\bar{q}})^T\bar{q} = \underbrace{\Xi(\hat{\bar{q}})^T\Xi(\hat{\bar{q}})}_{I_{3\times 3}}\delta\mathbf{q} + \underbrace{\Xi(\hat{\bar{q}})^T\hat{\bar{q}}}_{0}\delta q_4 \tag{5.62}$$

$$\delta\mathbf{q} = \Xi(\hat{\bar{q}})^T\bar{q} \tag{5.63}$$

$$\delta\mathbf{q} = \Xi(\hat{\bar{q}})^T(\Delta\bar{q} + \hat{\bar{q}}) \tag{5.64}$$

$$\delta\mathbf{q} = \Xi(\hat{\bar{q}})^T\Delta\bar{q} + \underbrace{\Xi(\hat{\bar{q}})^T\hat{\bar{q}}}_{0} \tag{5.65}$$

$$\delta\mathbf{q} = \Xi(\hat{\bar{q}})^T\Delta\bar{q} \tag{5.66}$$

The full quaternion state error can be recovered from the reduced representation using (3.12) and (5.53):

$$\delta\mathbf{q} = \Xi(\hat{\bar{q}})^T\Delta\bar{q} \tag{5.67}$$

$$\Xi(\hat{\bar{q}})\delta\mathbf{q} = \Xi(\hat{\bar{q}})\Xi(\hat{\bar{q}})^T\Delta\bar{q} \tag{5.68}$$

$$\Xi(\hat{\bar{q}})\delta\mathbf{q} = (I_{4\times 4} - \hat{\bar{q}}\hat{\bar{q}}^T)\Delta\bar{q} \tag{5.69}$$

$$\Xi(\hat{\bar{q}})\delta\mathbf{q} = \Delta\bar{q} - \hat{\bar{q}}\underbrace{\hat{\bar{q}}^T\Delta\bar{q}}_{\approx 0} \tag{5.70}$$

$$\Delta\bar{q} = \Xi(\hat{\bar{q}})\delta\mathbf{q} \tag{5.71}$$

Defining the deterministic matrix $\hat{\xi}$ below, the entire state error $\Delta\mathbf{x}$ can be converted to the reduced representation $\Delta\check{\mathbf{x}}$ and back, using (5.66) and (5.71):

$$\hat{\xi} = \begin{bmatrix} \Xi(\hat{\bar{q}}) & 0_{4\times 3} \\ 0_{3\times 3} & I_{3\times 3} \end{bmatrix} \tag{5.72}$$

$$\Delta\check{\mathbf{x}} = \hat{\xi}^T \Delta\mathbf{x} \tag{5.73}$$

$$\Delta\mathbf{x} = \hat{\xi}\Delta\check{\mathbf{x}} \tag{5.74}$$

The reduced representation of the covariance matrix can then be defined:

$$\begin{aligned}
\check{P} &= E(\Delta\check{\mathbf{x}}\Delta\check{\mathbf{x}}^T) && (5.75)\\
&= E(\hat{\xi}^T \Delta\mathbf{x}\Delta\mathbf{x}^T \hat{\xi}) && (5.76)\\
&= \hat{\xi}^T E(\Delta\mathbf{x}\Delta\mathbf{x}^T)\hat{\xi} && (5.77)\\
\check{P} &= \hat{\xi}^T P \hat{\xi} && (5.78)
\end{aligned}$$

The full covariance matrix is then recovered from the reduced representation:

$$\begin{aligned}
\Delta\check{\mathbf{x}}\Delta\check{\mathbf{x}}^T &= \hat{\xi}^T \Delta\mathbf{x}\Delta\mathbf{x}^T \hat{\xi} && (5.79)\\
\hat{\xi}\Delta\check{\mathbf{x}}\Delta\check{\mathbf{x}}^T\hat{\xi}^T &= \hat{\xi}\hat{\xi}^T \Delta\mathbf{x}\Delta\mathbf{x}^T \hat{\xi}\hat{\xi}^T && (5.80)\\
&= \left(I_{7\times7} - \begin{bmatrix} \hat{\bar{q}}\hat{\bar{q}}^T & 0_{4\times3} \\ 0_{3\times4} & 0_{3\times3} \end{bmatrix}\right) \Delta\mathbf{x}\Delta\mathbf{x}^T \left(I_{7\times7} - \begin{bmatrix} \hat{\bar{q}}\hat{\bar{q}}^T & 0_{4\times3} \\ 0_{3\times4} & 0_{3\times3} \end{bmatrix}\right) && (5.81)\\
&= \left(\Delta\mathbf{x} - \underbrace{\begin{bmatrix} \hat{\bar{q}}\hat{\bar{q}}^T & 0_{4\times3} \\ 0_{3\times4} & 0_{3\times3} \end{bmatrix}\Delta\mathbf{x}}_{\hat{\bar{q}}\hat{\bar{q}}^T\Delta\bar{q}\approx 0}\right)\left(\Delta\mathbf{x}^T - \underbrace{\Delta\mathbf{x}^T\begin{bmatrix} \hat{\bar{q}}\hat{\bar{q}}^T & 0_{4\times3} \\ 0_{3\times4} & 0_{3\times3} \end{bmatrix}}_{\Delta\bar{q}^T\hat{\bar{q}}\hat{\bar{q}}^T\approx 0}\right) && \\
& && (5.82)\\
\hat{\xi}\Delta\check{\mathbf{x}}\Delta\check{\mathbf{x}}^T\hat{\xi}^T &= \Delta\mathbf{x}\Delta\mathbf{x}^T && (5.83)\\
E(\hat{\xi}\Delta\check{\mathbf{x}}\Delta\check{\mathbf{x}}^T\hat{\xi}^T) &= E(\Delta\mathbf{x}\Delta\mathbf{x}^T) && (5.84)\\
\hat{\xi}\check{P}\hat{\xi}^T &= P && (5.85)
\end{aligned}$$

With the ability to easily convert from P to $\check{P}$ and back, the reduced covariance matrix can be inserted into the EKF framework. Conceptually, tracking the covariance as $\check{P}$ and converting to P whenever it is needed, then reducing back to $\check{P}$

for storage gives the following in all EKF equations that use the covariance matrix:

Predict

$$\check{P}_k^- = \hat{\xi}^T (F_k \hat{\xi} \check{P}_{k-1}^+ \hat{\xi}^T F_k^T + G_k Q_k G_k^T) \hat{\xi} \tag{5.86}$$

$$= \hat{\xi}^T F_k \hat{\xi} \check{P}_{k-1}^+ \hat{\xi}^T F_k^T \hat{\xi} + \hat{\xi}^T G_k Q_k G_k^T \hat{\xi} \tag{5.87}$$

Update

$$L_k = \hat{\xi} \check{P}_k^- \hat{\xi}^T H_k^T (H_k \hat{\xi} \check{P}_k^- \hat{\xi}^T H_k^T + R_k)^{-1} \tag{5.88}$$

$$\check{P}_k^+ = \check{P}_k^- - \hat{\xi}^T L_k H_k \hat{\xi} \check{P}_k^- \hat{\xi}^T \hat{\xi} \tag{5.89}$$

Then by defining:

$$\check{F} = \hat{\xi}^T F \hat{\xi} \tag{5.90}$$

$$\check{G} = \hat{\xi}^T G \tag{5.91}$$

$$\check{H} = H \hat{\xi} \tag{5.92}$$

$$\check{L} = \hat{\xi}^T L \tag{5.93}$$

$\check{F}$ can be simplified (3.11):

$$\check{F} = \begin{bmatrix} \Xi(\hat{\bar{q}})^T & 0_{3\times3} \\ 0_{4\times3} & I_{3\times3} \end{bmatrix} \begin{bmatrix} \mathcal{L}(\bar{q}(\hat{\boldsymbol{\omega}}_k)) & -\frac{1}{2}\Xi(\bar{q}_{k-1})\Delta_t \\ 0_{3\times4} & I_{3\times3} \end{bmatrix} \begin{bmatrix} \Xi(\hat{\bar{q}}) & 0_{4\times3} \\ 0_{3\times3} & I_{3\times3} \end{bmatrix} \tag{5.94}$$

$$= \begin{bmatrix} \Xi(\hat{\bar{q}})^T \mathcal{L}(\bar{q}(\hat{\boldsymbol{\omega}}_k)) \Xi(\hat{\bar{q}}) & -\frac{1}{2}\Xi(\hat{\bar{q}})^T \Xi(\hat{\bar{q}})\Delta_t \\ 0_{3\times4} & I_{3\times3} \end{bmatrix} \tag{5.95}$$

$$= \begin{bmatrix} \Lambda & -\frac{1}{2}\Delta_t I_{3\times3} \\ 0_{3\times4} & I_{3\times3} \end{bmatrix} \tag{5.96}$$

Where Λ (derived in the appendix) is:

$$\Lambda = \begin{bmatrix} 1 & \frac{1}{2}\hat{\omega}_3 & -\frac{1}{2}\hat{\omega}_2 \\ -\frac{1}{2}\hat{\omega}_3 & 1 & \frac{1}{2}\hat{\omega}_1 \\ \frac{1}{2}\hat{\omega}_2 & -\frac{1}{2}\hat{\omega}_1 & 1 \end{bmatrix} \tag{5.97}$$

Likewise $\check{G}$ can also be simplified:

$$\check{G} = \begin{bmatrix} \Xi(\hat{\bar{q}})^T & 0_{3\times3} \\ 0_{4\times3} & I_{3\times3} \end{bmatrix} \begin{bmatrix} -\frac{1}{2}\Xi(\bar{q}) & 0_{4\times3} \\ 0_{3\times3} & I_{3\times3} \end{bmatrix} \tag{5.98}$$

$$= \begin{bmatrix} -\frac{1}{2}I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} \end{bmatrix} \tag{5.99}$$

Making the replacements, the entire EKF can then be run without ever computing the full covariance matrix P. The final form of the Quaternion EKF is then[4]:

Predict

$$\hat{\mathbf{x}}_k^- = F_k \hat{\mathbf{x}}_{k-1}^+ \tag{5.100}$$

$$\check{P}_k^- = \check{F}_k \check{P}_{k-1}^+ \check{F}_k^T + \check{G}_k Q \check{G}_k^T \tag{5.101}$$

Update

$$\mathbf{z}_k = \tilde{\mathbf{y}}_k - \begin{pmatrix} C_{n\to b}(\hat{\bar{q}}_k^-)\mathbf{m}^n \\ C_{n\to b}(\hat{\bar{q}}_k^-)\mathbf{g}^n \end{pmatrix} \tag{5.102}$$

$$\check{H}_k = \begin{bmatrix} H_k^m \\ H_k^a \end{bmatrix} \hat{\xi} \tag{5.103}$$

$$\check{L}_k = \check{P}_k^- \check{H}_k^T (\check{H}_k \check{P}_k^- \check{H}_k^T + R)^{-1} \tag{5.104}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \hat{\xi} \check{L}_k \mathbf{z}_k \tag{5.105}$$

$$\check{P}_k^+ = \check{P}_k^- - \check{L}_k \check{H}_k \check{P}_k^- \tag{5.106}$$

As the EKF does not model acceleration-induced bias, gyroscope scale factor, or any alignment errors, the simulation is first run with those error sources removed in order to verify the filter properly tracks the attitude and gyroscope bias. Covariance matrices are given in the appendix. Figures 5.2, 5.3, and 5.4 show that the EKF

[4] Values used in the simulation for $\check{P}_0$, Q, and R can be found in the appendix

tracks the attitude of the vehicle accurately through all the maneuvers, and figures 5.5, 5.6, and 5.7 show the bias estimate of the gyroscope follows the plant very closely.

At approximately 264 and 273 seconds in the simulation, the aircraft passes through the two singularities of the Tait-Bryan angle representation (straight up and straight down respectively). While this is not a problem for the quaternion representation the state uses, it does cause issues when attempting to plot the representation using Tait-Bryan angles; both the heading and roll plots diverge in this region. Figure 5.8 shows the underlying quaternion components of the state and the state estimate, where it is clear there are no issues with the quaternion state in either of these regions.

After reintroducing the scale factor errors in the instruments and using the same covariance matrices and random number draws, the effect can clearly be seen (figures 5.9 through 5.14). The gyroscope bias estimate is more accurately a gyroscope error estimate; the introduction of acceleration bias and gyroscope scale factor errors results in an estimate that includes all three. The bias estimate shows the EKF tracking the actual bias plus a near-constant offset (normal force-induced bias) with significant changes in the offset during rotations (scale factor error).

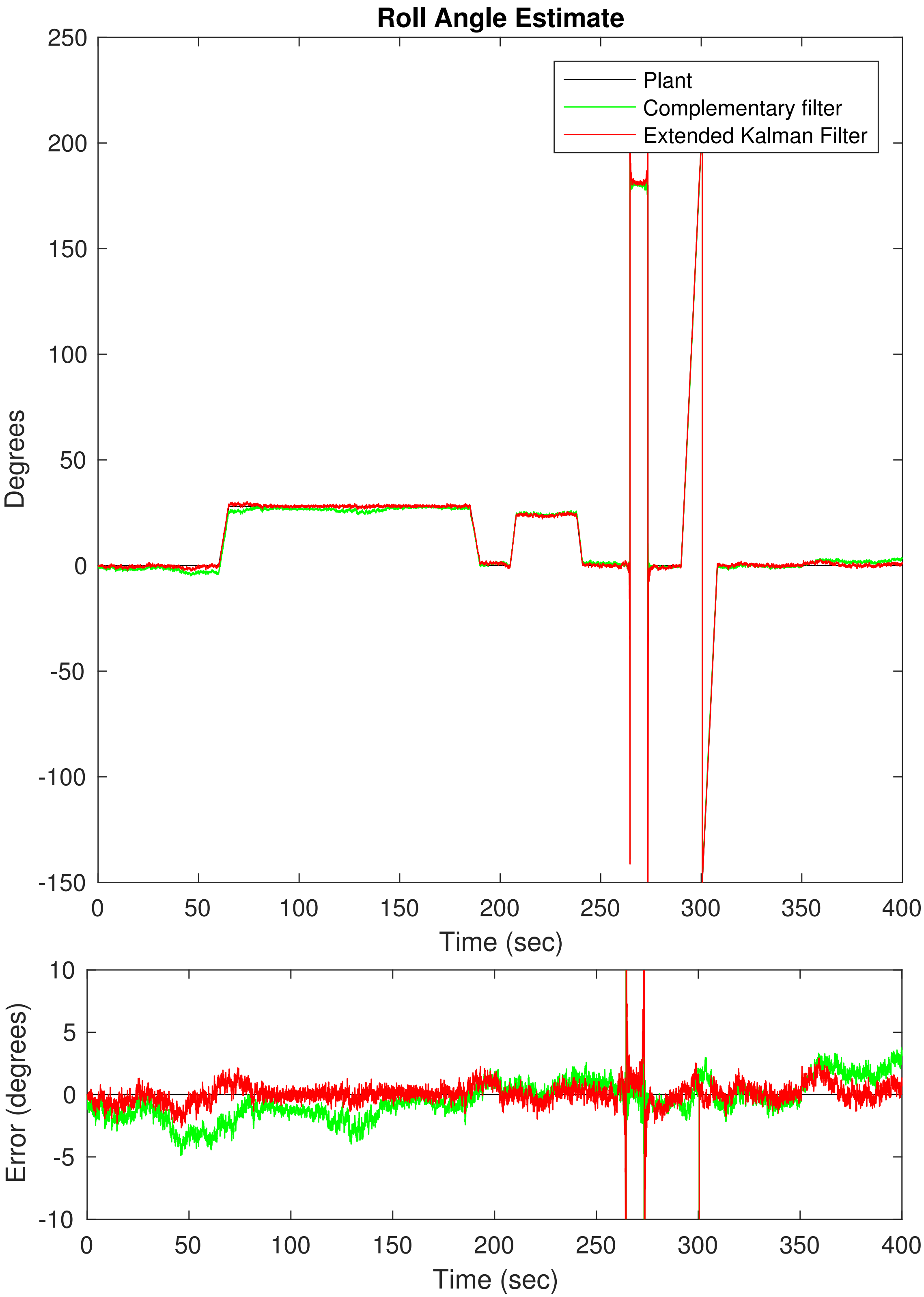

Figure 5.2: Extended Kalman filter roll estimate with no scale factor errors

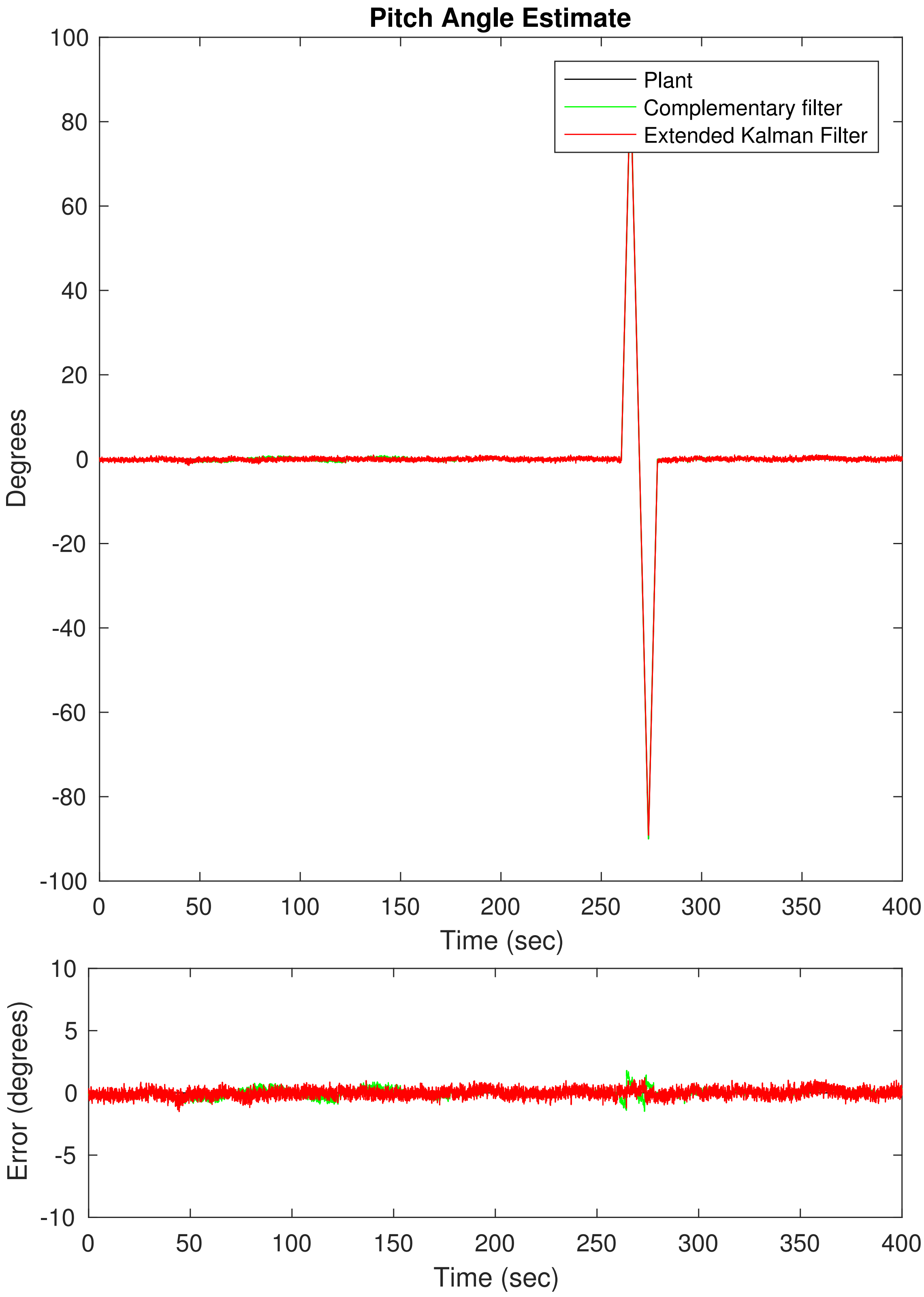

Figure 5.3: Extended Kalman filter pitch estimate with no scale factor errors

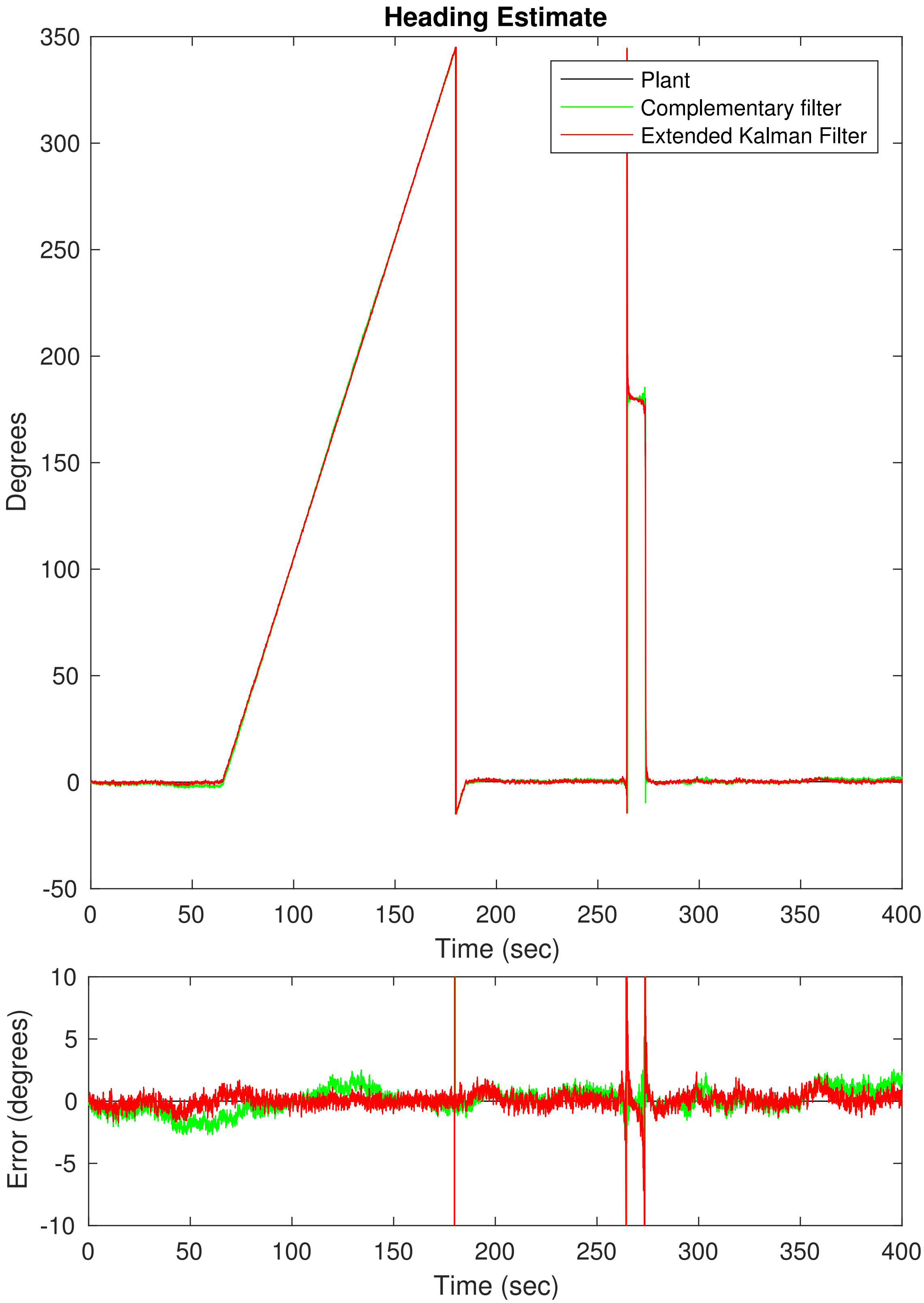

Figure 5.4: Extended Kalman filter heading estimate with no scale factor errors

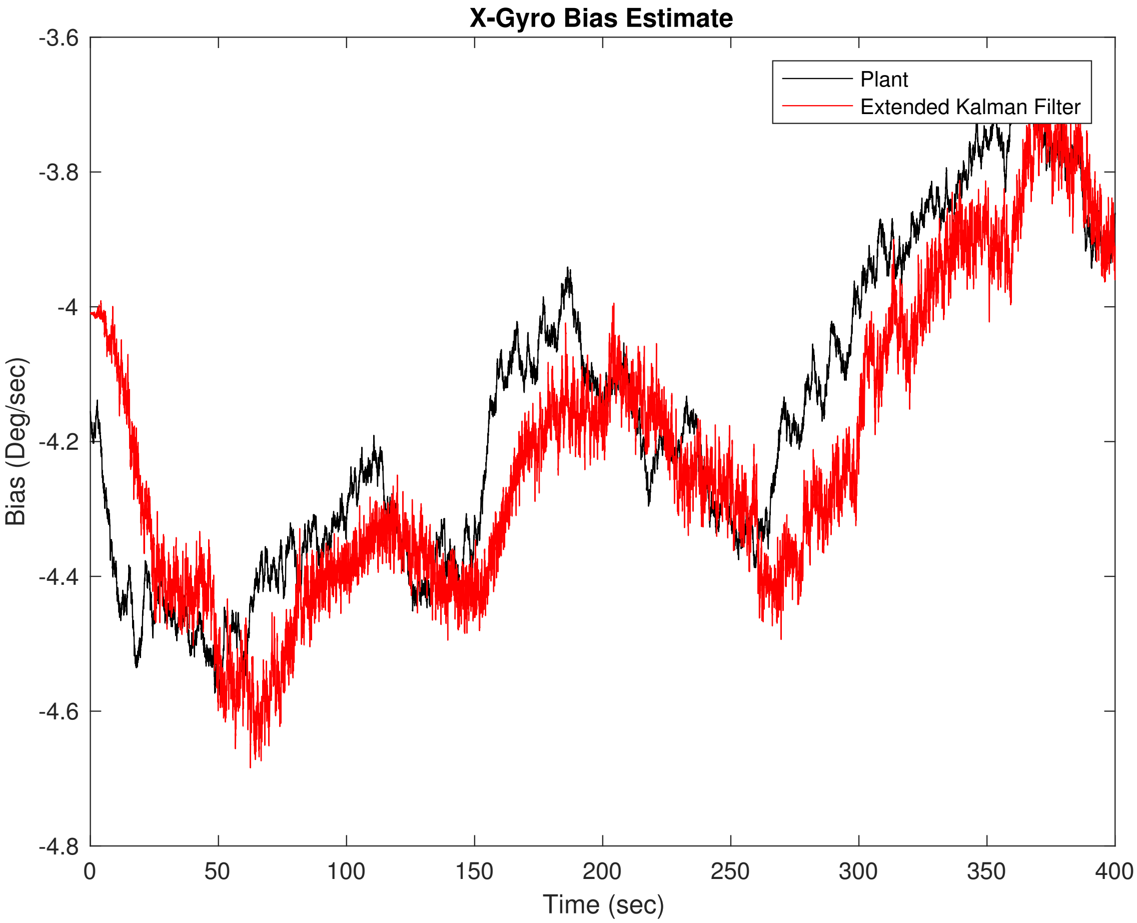

Figure 5.5: Extended Kalman filter x-gyroscope bias estimate with no scale factor errors

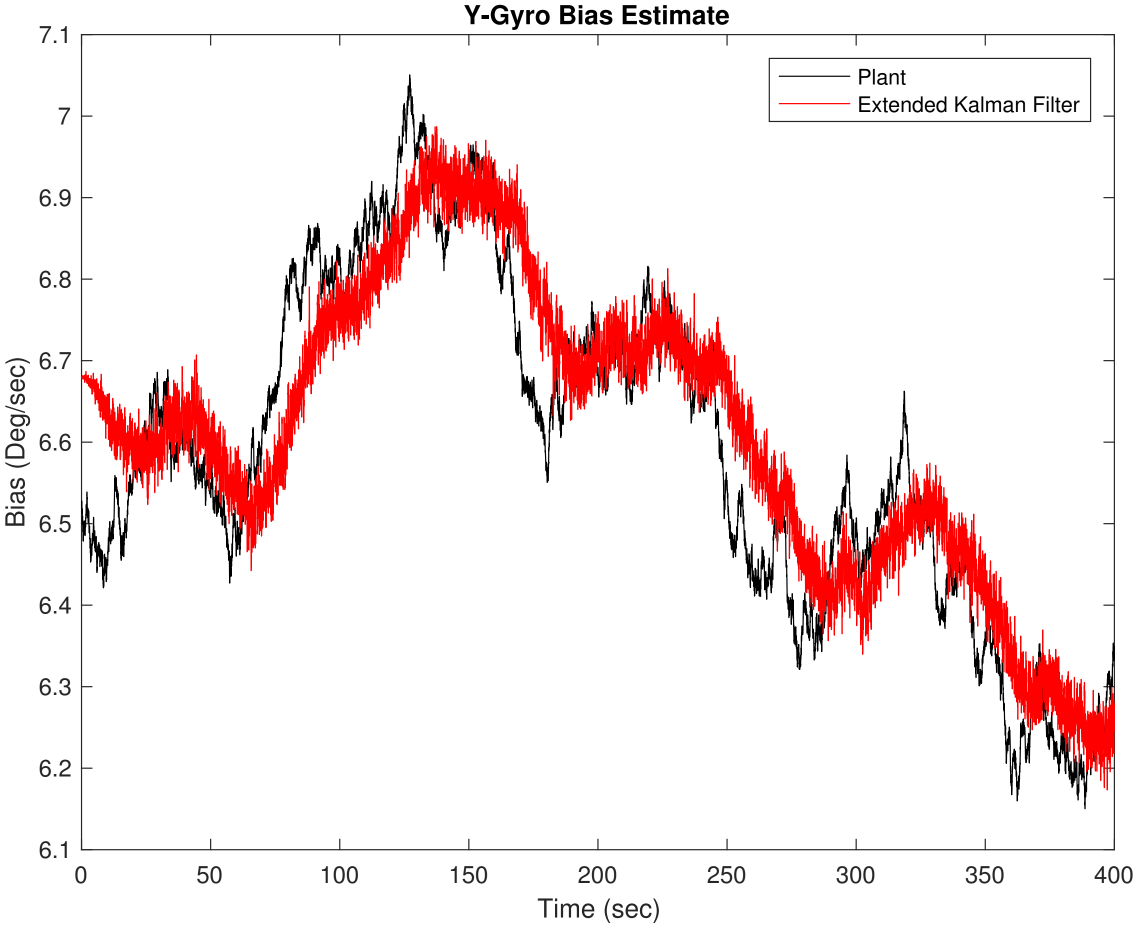

Figure 5.6: Extended Kalman filter y-gyroscope bias estimate with no scale factor errors

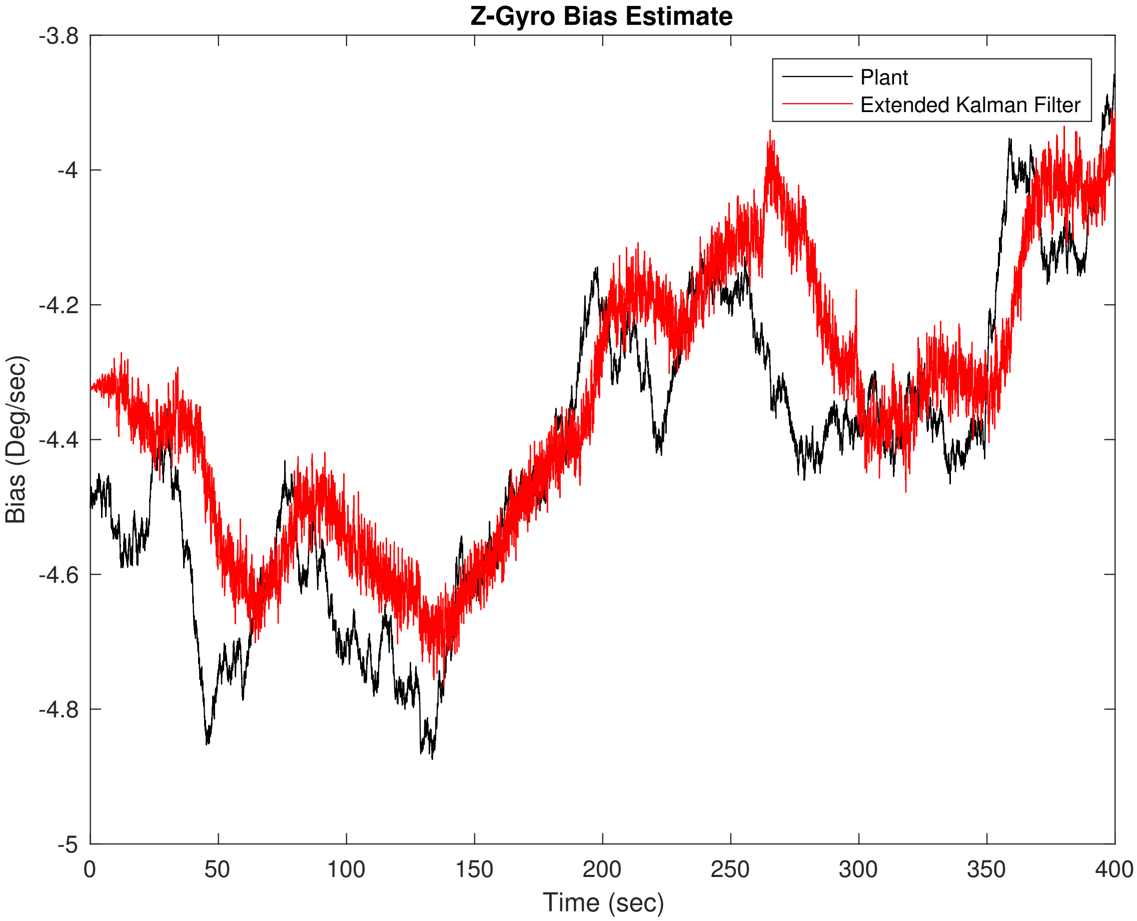

Figure 5.7: Extended Kalman filter z-gyroscope bias estimate with no scale factor errors

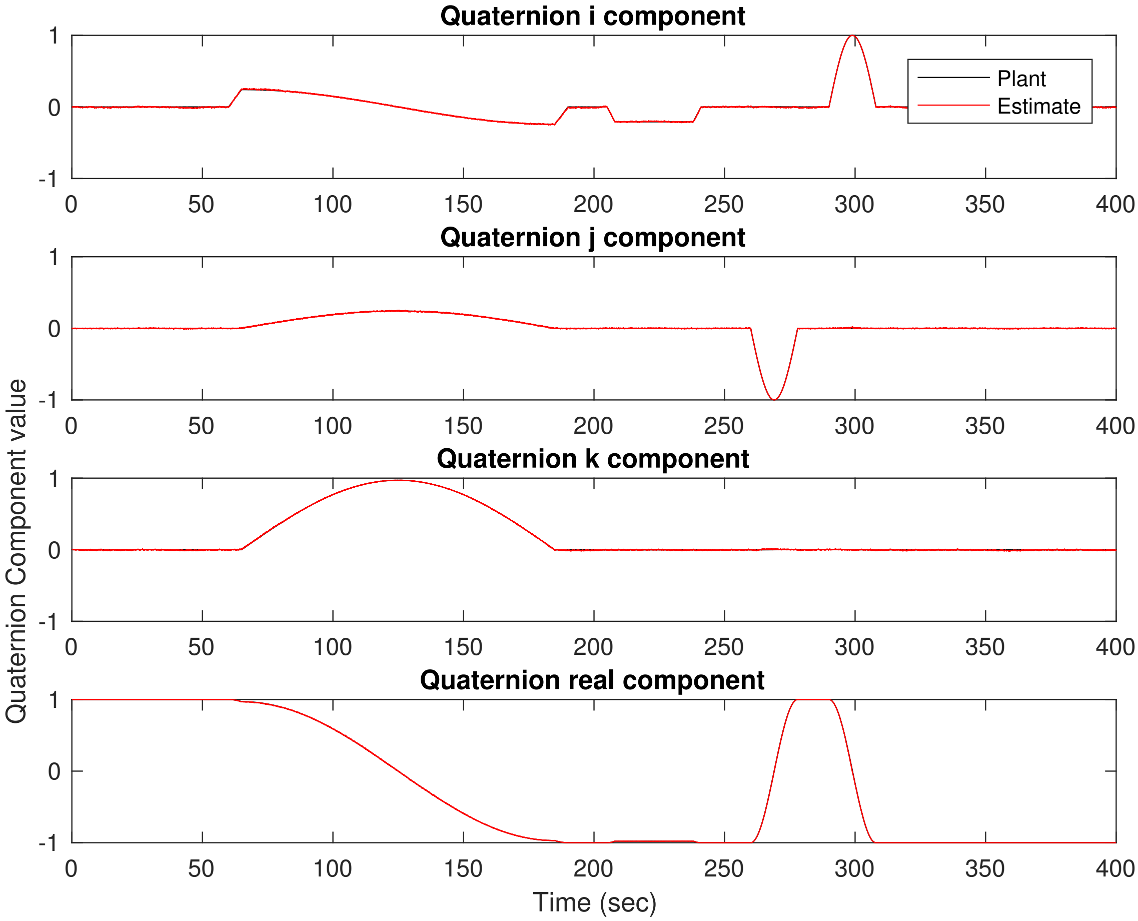

Figure 5.8: State quaternion components with no scale factor errors

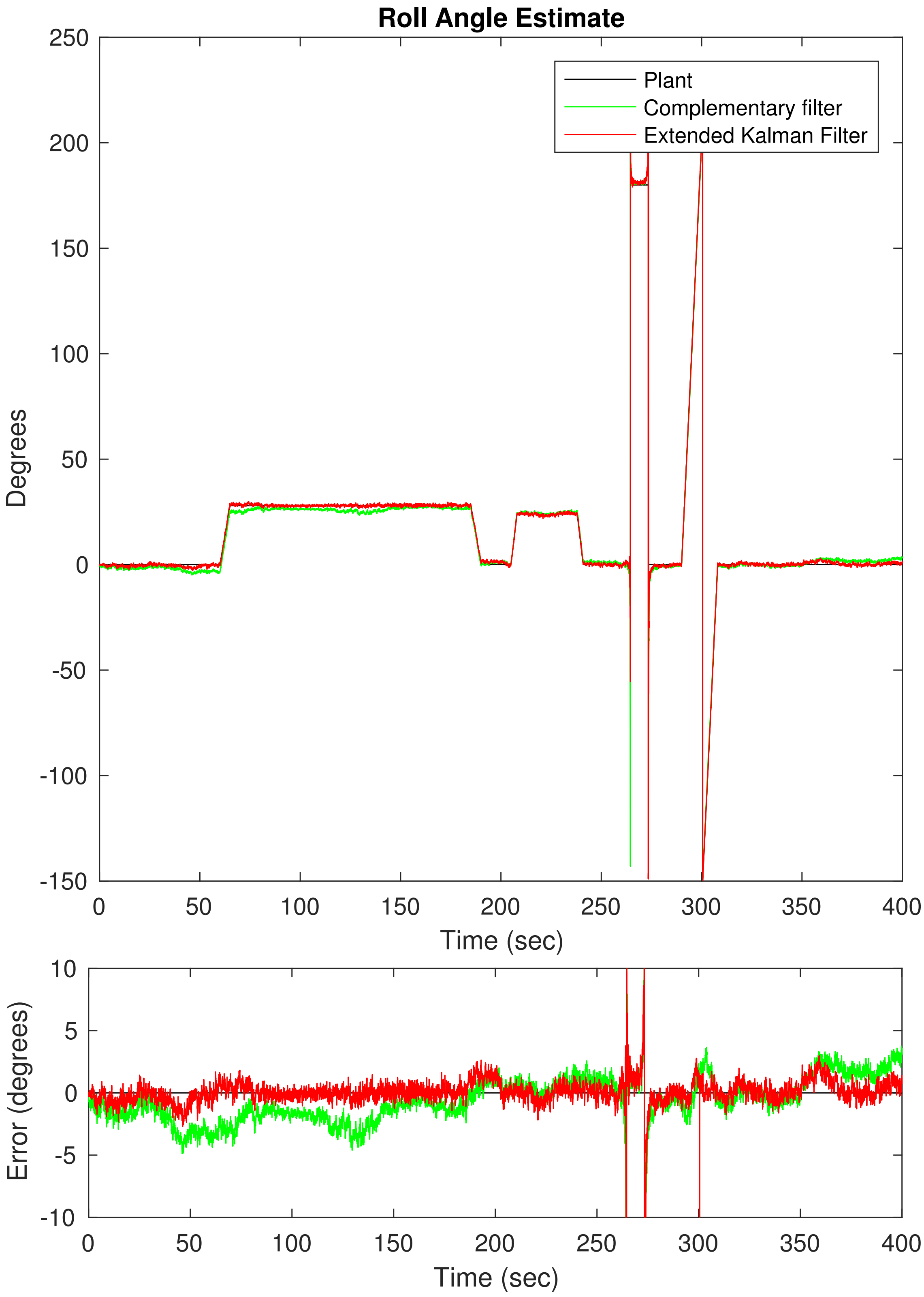

Figure 5.9: Extended Kalman filter roll estimate with scale factor errors

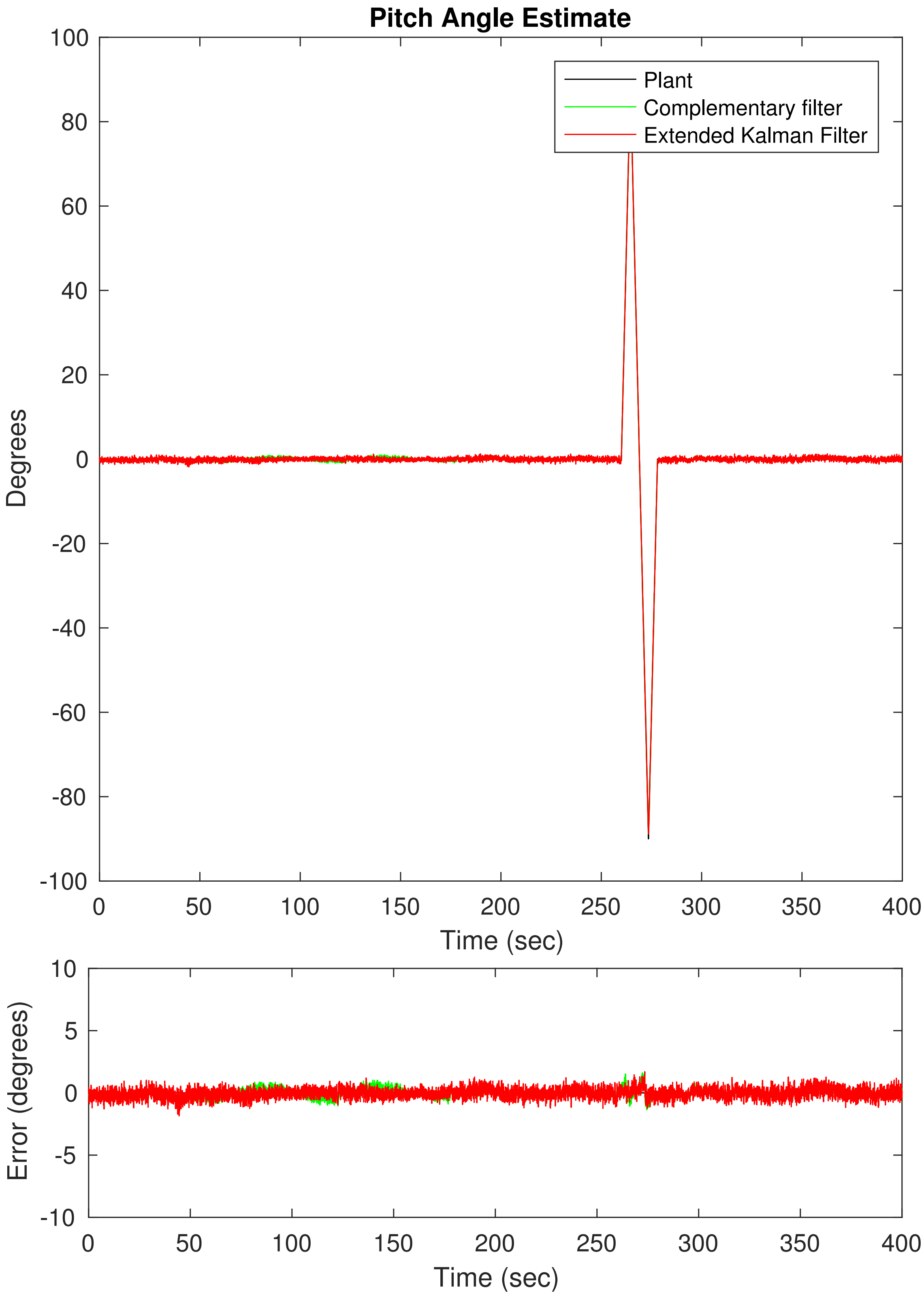

Figure 5.10: Extended Kalman filter pitch estimate with scale factor errors

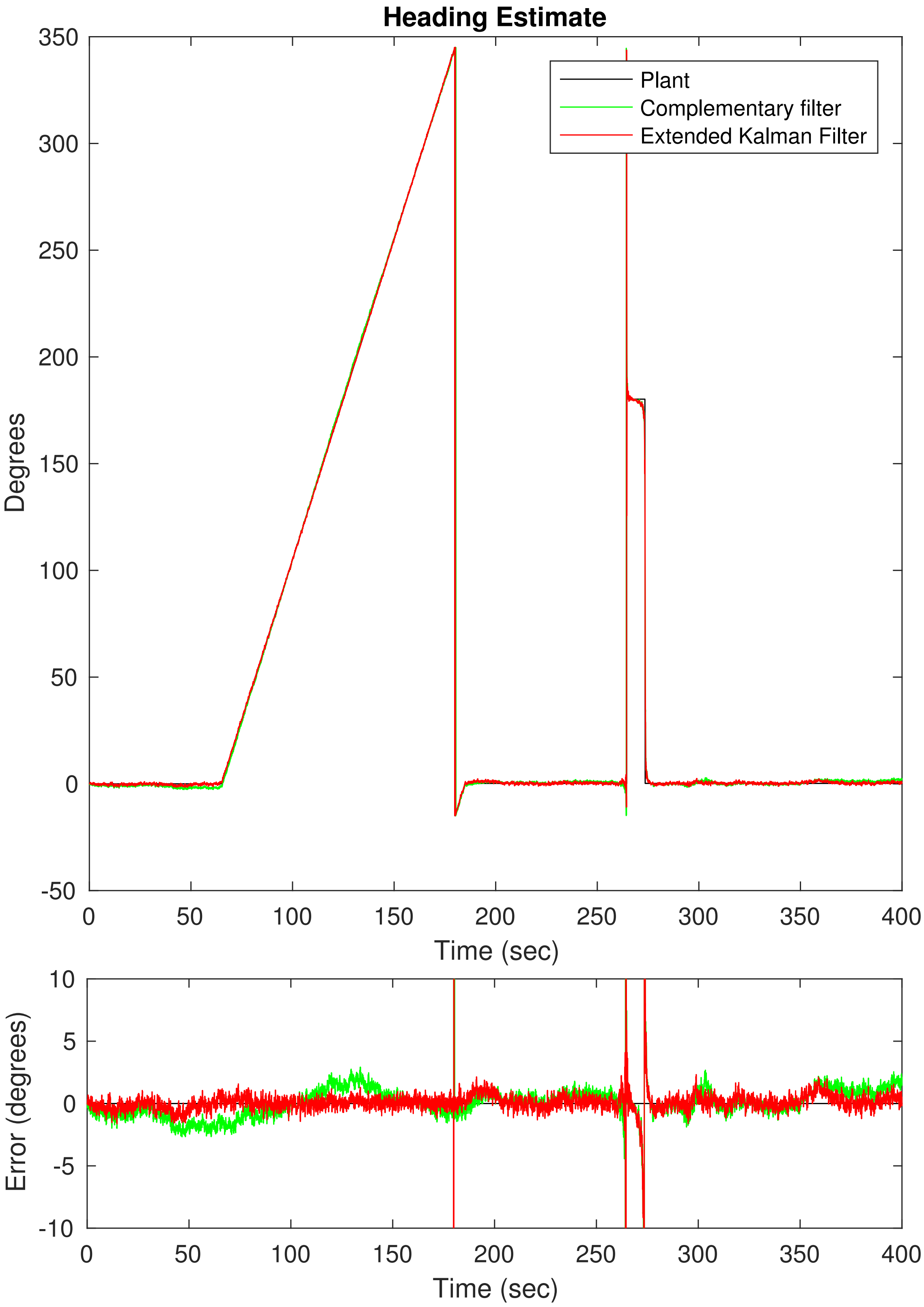

Figure 5.11: Extended Kalman filter heading estimate with scale factor errors

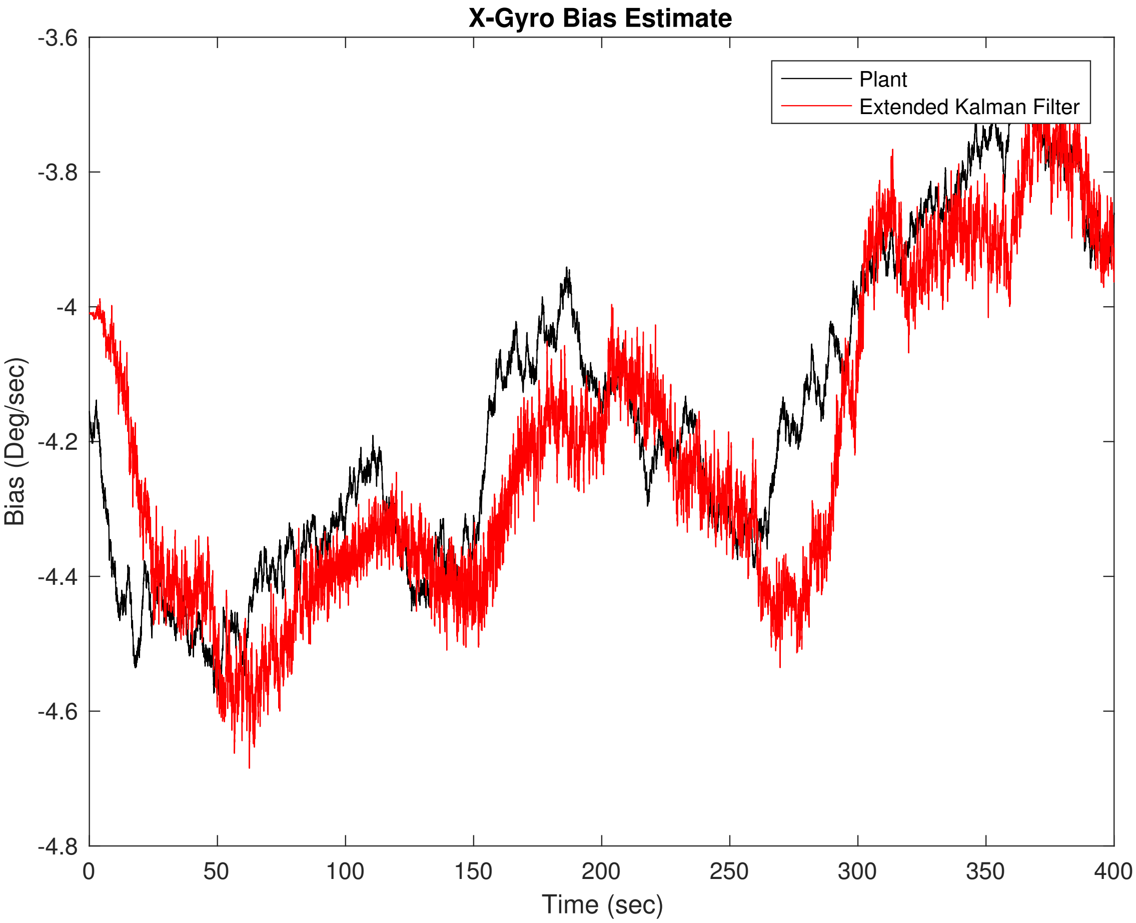

Figure 5.12: Extended Kalman filter x-gyroscope bias estimate with scale factor errors

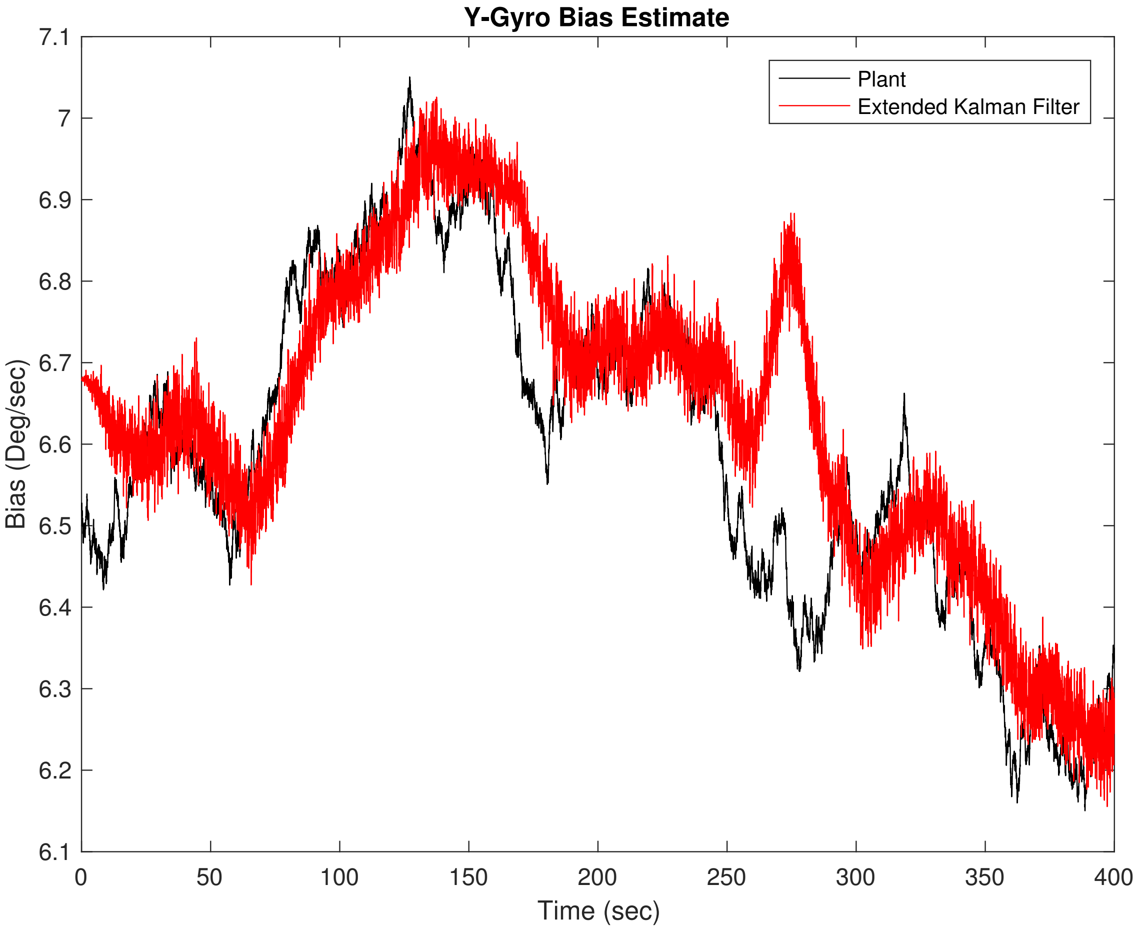

Figure 5.13: Extended Kalman filter y-gyroscope bias estimate with scale factor errors

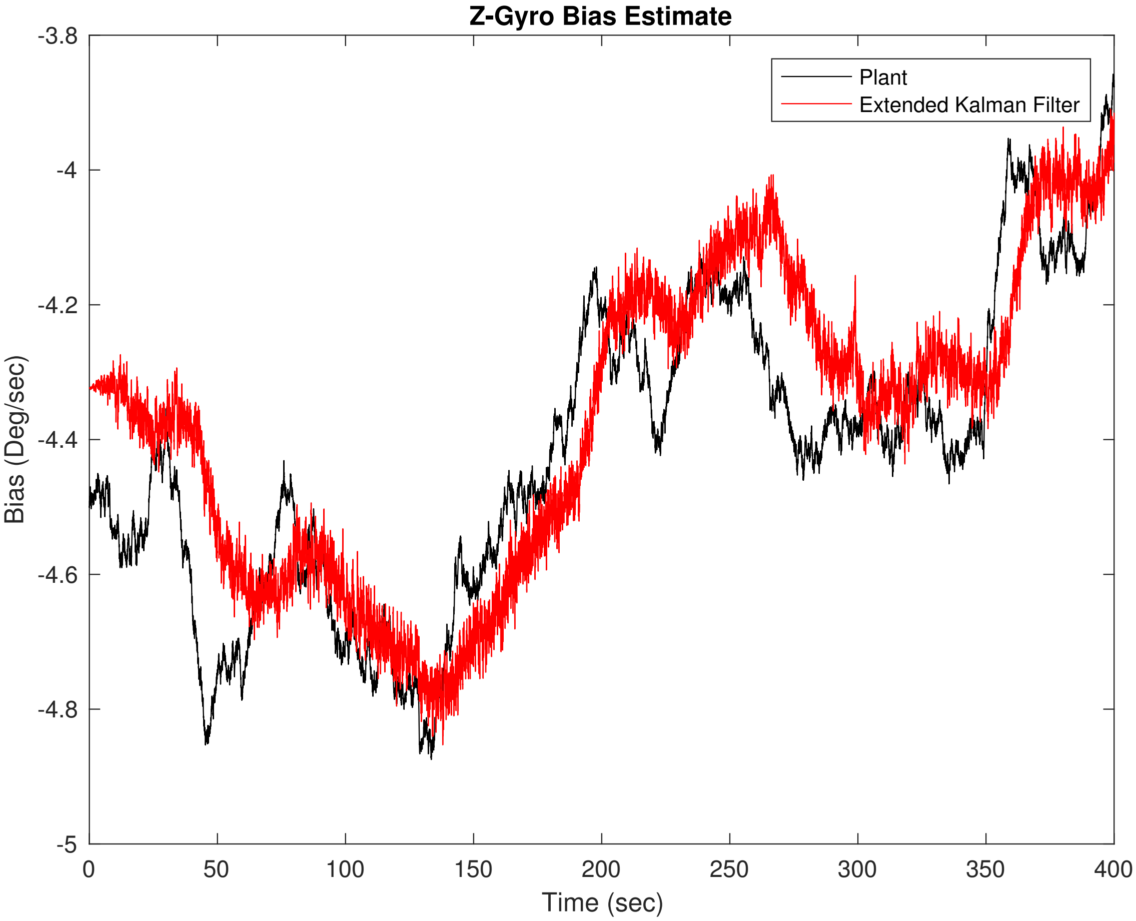

Figure 5.14: Extended Kalman filter z-gyroscope bias estimate with scale factor errors

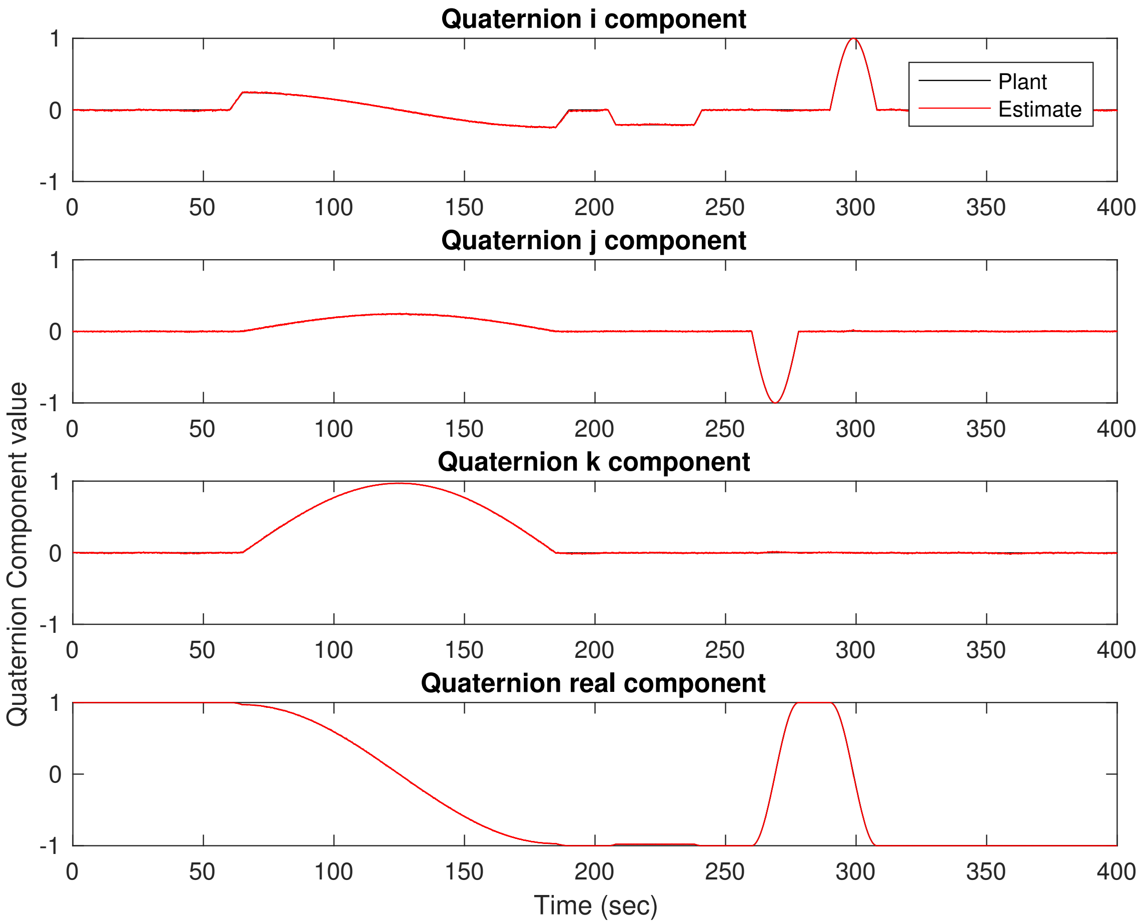

Figure 5.15: State quaternion components with scale factor errors

Chapter 6

Filter Comparison

In order to truly compare the performance of the two filters, one set of random numbers is not sufficient. A Monte Carlo of twenty-four runs each was conducted, and the maximum and minimum values at each timestep for all of the runs was plotted. All of the sensor errors (bias, scale factor, AWGN) were pulled from a different set of random numbers on each run, and both the EKF and the complementary filter used the same sets.

Figures 6.1, 6.2, and 6.3 show the performance of the two filters in roll, pitch, and heading respectively. Upon inspection, it becomes clear that pitch is measured most accurately, followed by heading and then roll. Pitch is simplest to calculate because it is directly observable at all times by the gravity vector, and it is observable at some times by the magnetic field vector. Heading becomes slightly more difficult, as it is only observable by the magnetic field vector. Roll is the most troublesome, as it is only indirectly observable by making some reasonable assumptions and then combining multiple noisy sensors to obtain an estimate, which is sensitive to sensor errors.

Through each of the three figures, it is clear that the EKF performed significantly better than the complementary filter. While the complementary filter loses accuracy over time as the gyroscope drifts, the EKF retains relatively high accuracy throughout the run, with only minor short-term errors due to aggressive maneuvering.

In order to demonstrate robustness of the system, a second Monte Carlo was conducted with incorrect initial conditions given to the filters: the heading and roll were both 180° off, pitch was 45° off, and the gyroscope bias estimate was set to 0. This is shown in figures 6.4, 6.5, and 6.6. Clearly, the complementary filter in this form is useless (due to the gyroscope bias error), while the EKF converges within about one minute. While there are other methods of dealing with drifting gyroscopes for a complementary filter, this certainly underscores the importance of estimating gyroscope drift in attitude estimation.

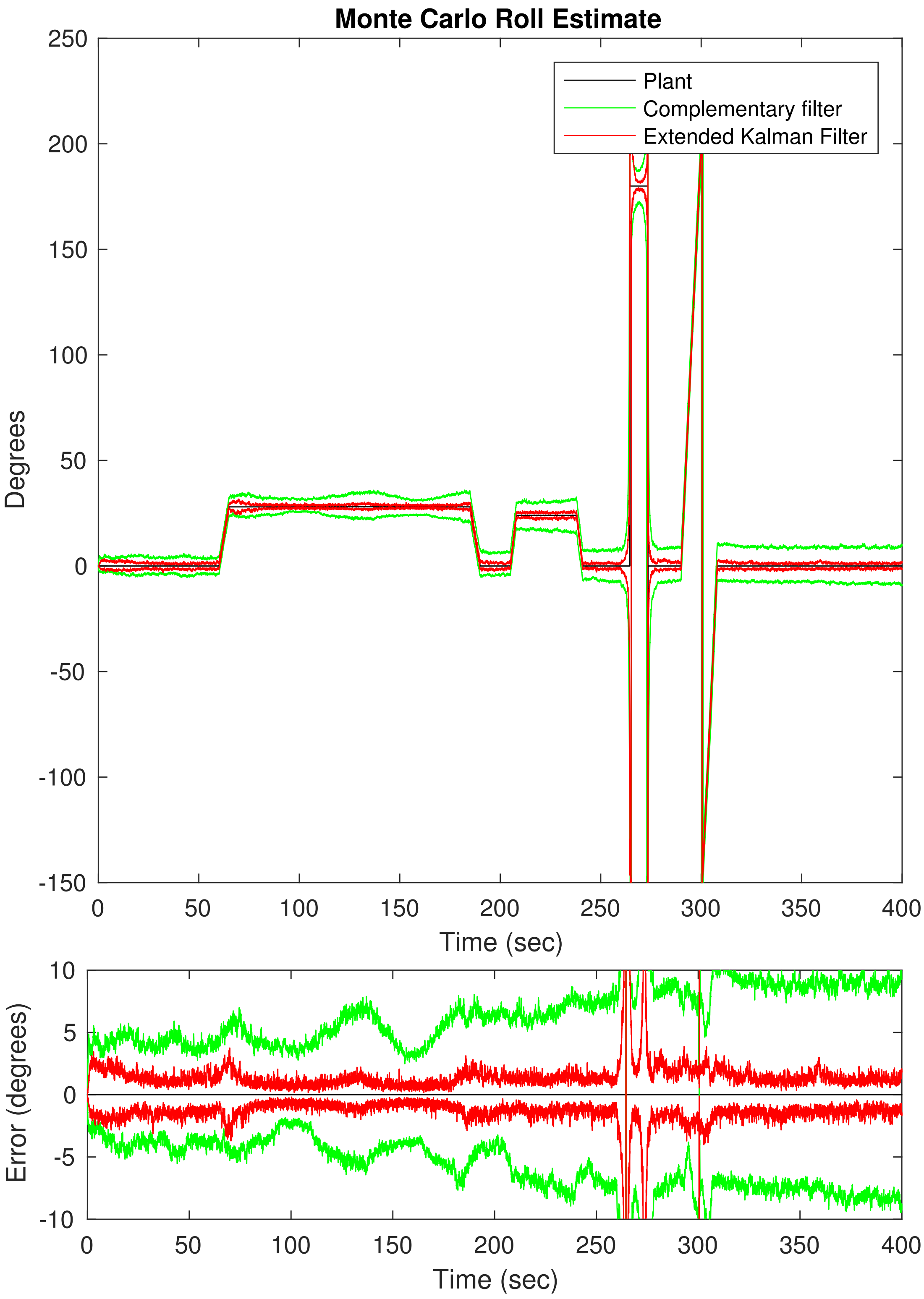

Figure 6.1: Monte Carlo run comparing complementary to EKF roll estimate

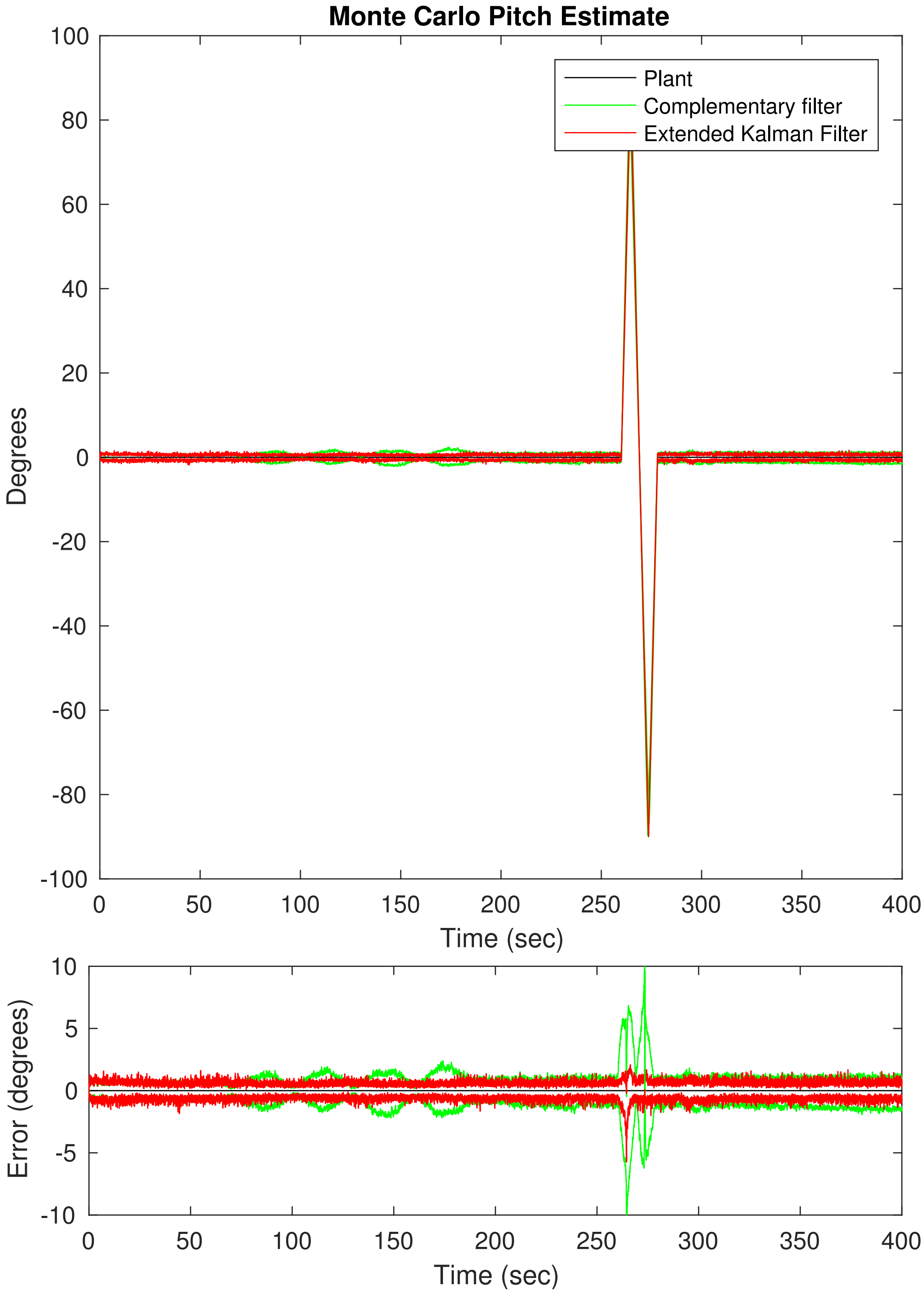

Figure 6.2: Monte Carlo run comparing complementary to EKF pitch estimate

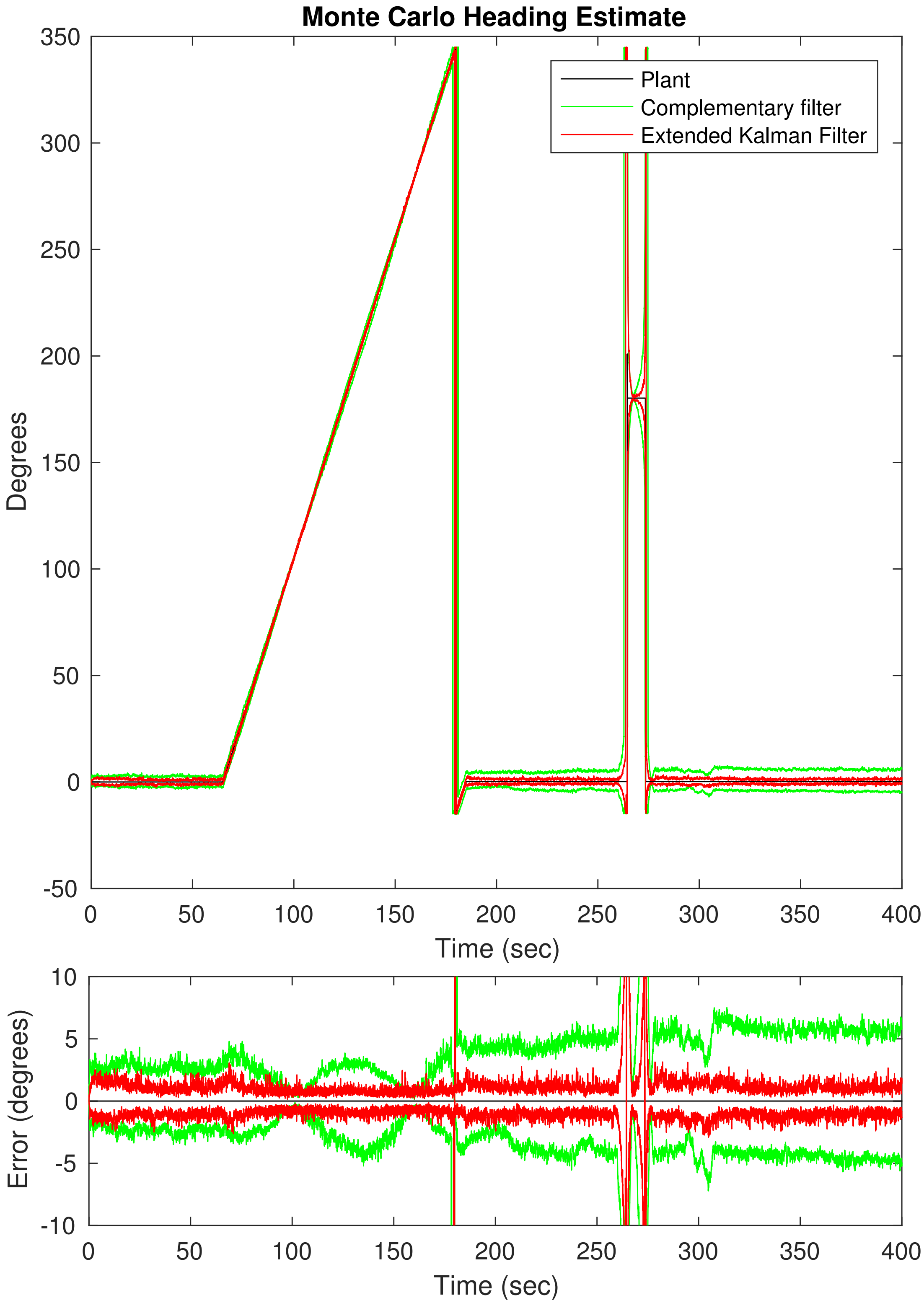

Figure 6.3: Monte Carlo run comparing complementary to EKF heading estimate

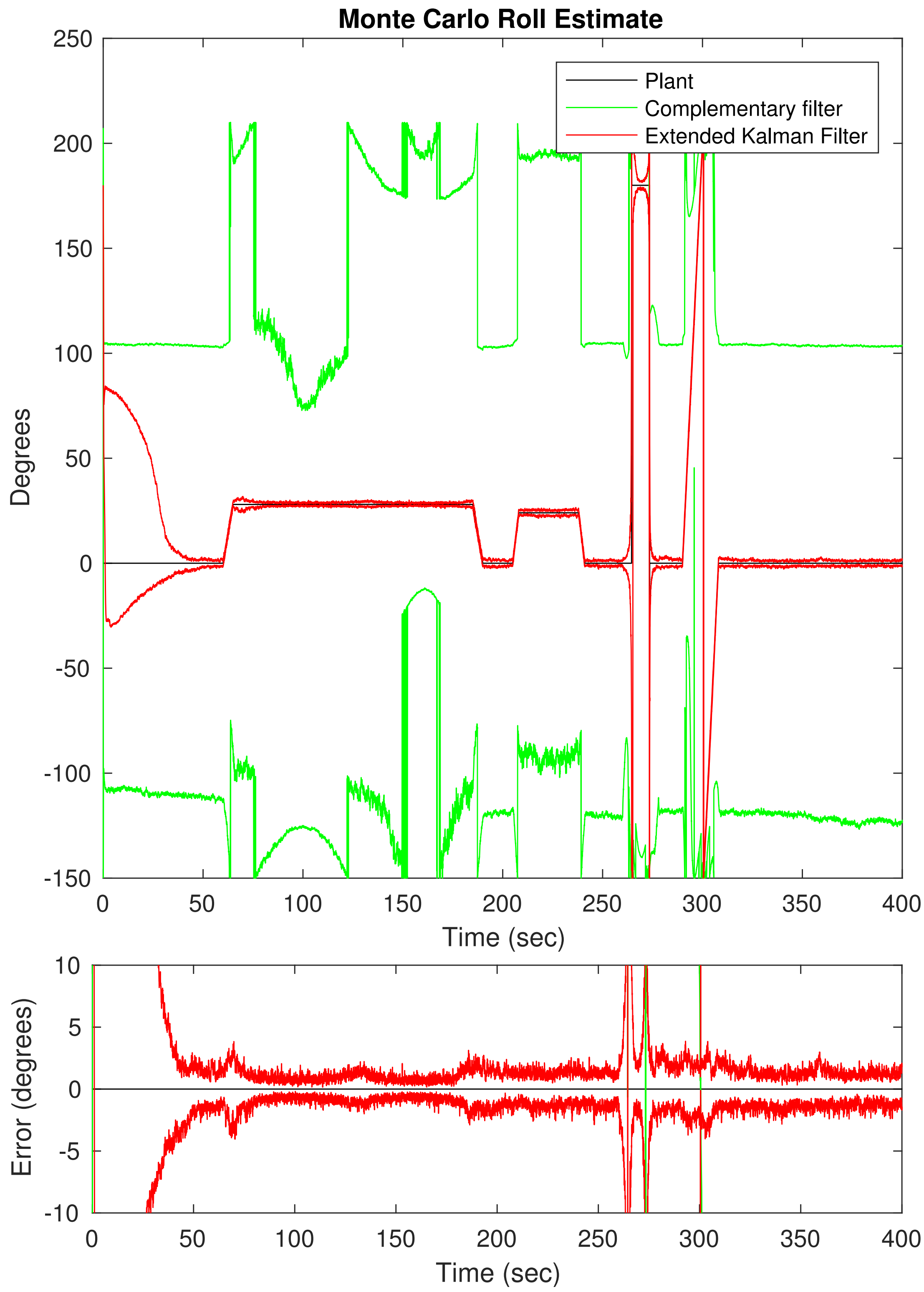

Figure 6.4: Monte Carlo run comparing complementary to EKF roll estimate with incorrect initial conditions

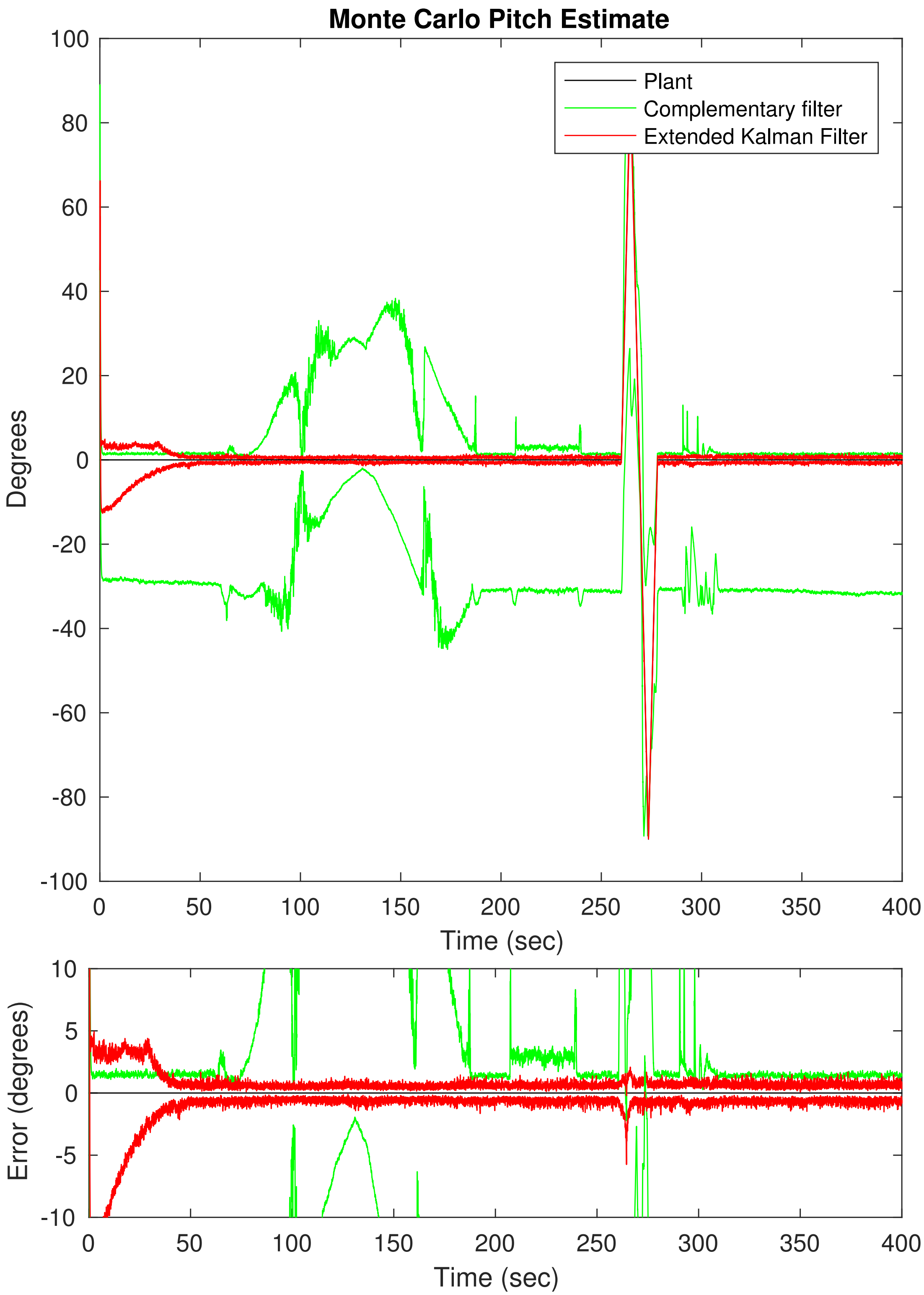

Figure 6.5: Monte Carlo run comparing complementary to EKF pitch estimate with incorrect initial conditions

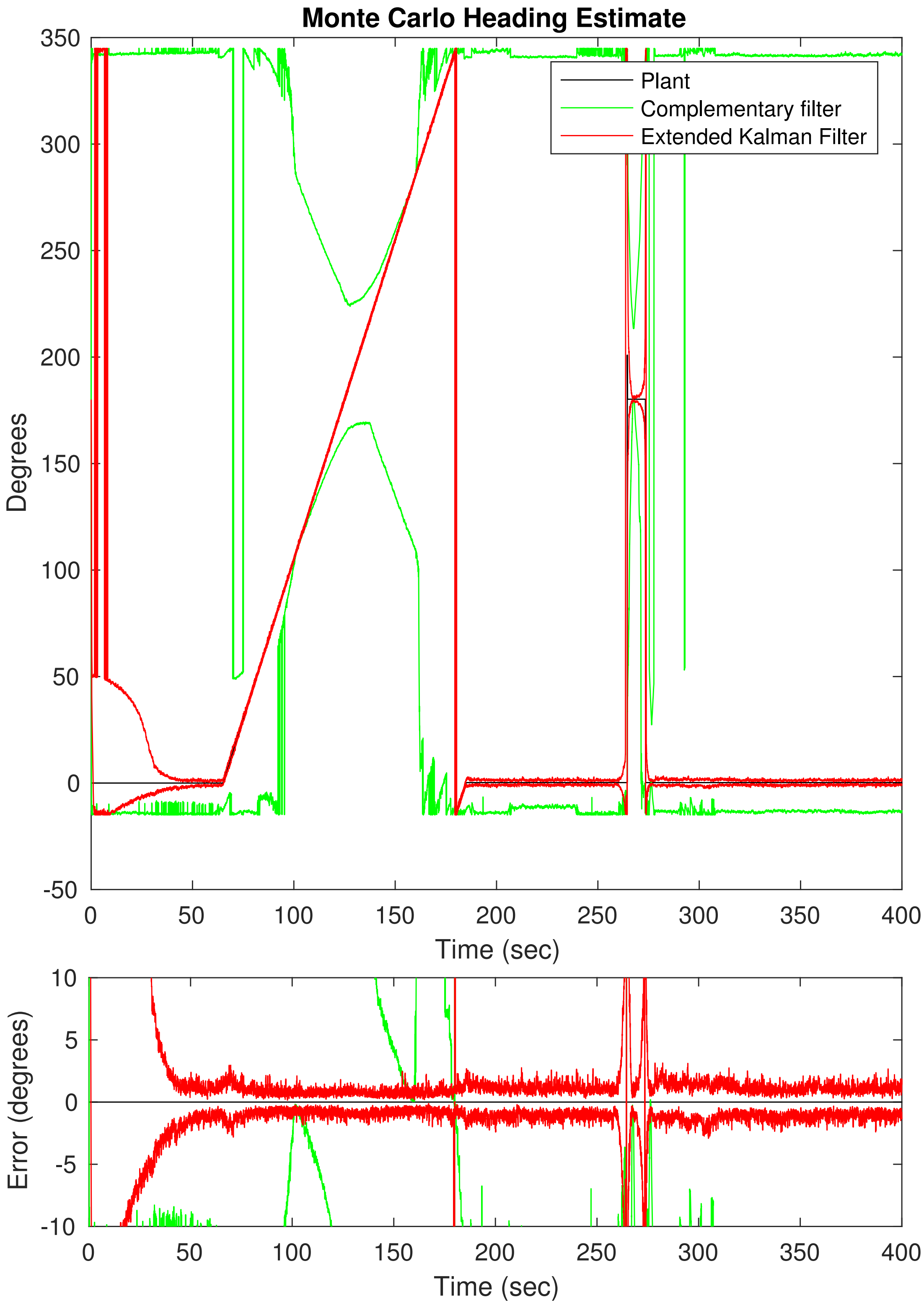

Figure 6.6: Monte Carlo run comparing complementary to EKF heading estimate with incorrect initial conditions

Chapter 7

Conclusion

Both the complementary filter and the EKF are powerful tools for aircraft attitude estimation. The simplicity of the complementary filter makes it a very attractive option despite its shortcomings. The primary drawback of the complementary filter in this implementation was the lack of gyroscope bias estimation; for short flights (e.g. remote control airplane), or with the incorporation of other techniques for tracking bias, the complementary filter could be used. However, the flexibility and power of the EKF makes it the default option for most commercial and military grade attitude and heading reference systems; the ability to incorporate the statistics of the process noise and the measurements into the state estimate makes it a very capable tool.

Appendix

All of the sensors are modeled with errors pulled from the following Gaussian distributions (see Chapter 1):

Accelerometer:

$$S_a = \mathcal{N}\left(\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}, \begin{bmatrix}.01 & .0009 & .0009\\ .0009 & .01 & .0009\\ .0009 & .0009 & .01\end{bmatrix}\right) \tag{A.1}$$

$$\mathbf{b}_a = \mathcal{N}\left(\begin{pmatrix}0\\0\\0\end{pmatrix}, \begin{pmatrix}1\\1\\1\end{pmatrix}\right) m/s^2 \tag{A.2}$$

$$\mathbf{w}_a = \mathcal{N}\left(\begin{pmatrix}0\\0\\0\end{pmatrix}, \begin{pmatrix}.0278\\.0278\\.0278\end{pmatrix}\right) m/s^2 \tag{A.3}$$

Magnetometer:

$$S_m = \mathcal{N}\left(\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}, \begin{bmatrix}.09 & 2.5\text{e}{-7} & 2.5\text{e}{-7}\\ 2.5\text{e}{-7} & .09 & 2.5\text{e}{-7}\\ 2.5\text{e}{-7} & 2.5\text{e}{-7} & .09\end{bmatrix}\right) \tag{A.4}$$

$$\mathbf{b}_m = \mathcal{N}\left(\begin{pmatrix}0\\0\\0\end{pmatrix}, \begin{pmatrix}.0025\\.0025\\.0025\end{pmatrix}\right) G \tag{A.5}$$

$$\mathbf{w}_m = \mathcal{N}\left(\begin{pmatrix}0\\0\\0\end{pmatrix}, \begin{pmatrix}1\text{e}{-4}\\1\text{e}{-4}\\1\text{e}{-4}\end{pmatrix}\right) G \tag{A.6}$$

Gyroscope:

$$S_\omega = \mathcal{N}\left(\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}, \begin{bmatrix}6.25\text{e}{-4} & 6.25\text{e}{-6} & 6.25\text{e}{-6}\\ 6.25\text{e}{-6} & 6.25\text{e}{-4} & 6.25\text{e}{-6}\\ 6.25\text{e}{-6} & 6.25\text{e}{-6} & 6.25\text{e}{-4}\end{bmatrix}\right) \tag{A.7}$$

$$A = \mathcal{N}\left(\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}, \begin{bmatrix}2.5\text{e}{-7} & 2.5\text{e}{-7} & 2.5\text{e}{-7}\\ 2.5\text{e}{-7} & 2.5\text{e}{-7} & 2.5\text{e}{-7}\\ 2.5\text{e}{-7} & 2.5\text{e}{-7} & 2.5\text{e}{-7}\end{bmatrix}\right) \tag{A.8}$$

$$\mathbf{b}_\omega = \mathcal{N}\left(\begin{pmatrix}0\\ 0\\ 0\end{pmatrix}, \begin{pmatrix}.01\\ .01\\ .01\end{pmatrix}\right) rad/s \tag{A.9}$$

$$\mathbf{w}_\omega = \mathcal{N}\left(\begin{pmatrix}0\\ 0\\ 0\end{pmatrix}, \begin{pmatrix}2.5\text{e}{-5}\\ 2.5\text{e}{-5}\\ 2.5\text{e}{-5}\end{pmatrix}\right) rad/s \tag{A.10}$$

$$\mathbf{w}_{\omega d} = \mathcal{N}\left(\begin{pmatrix}0\\ 0\\ 0\end{pmatrix}, \begin{pmatrix}2.5\text{e}{-9}\\ 2.5\text{e}{-9}\\ 2.5\text{e}{-9}\end{pmatrix}\right) rad/s^2 \tag{A.11}$$

The EKF (Chapter 5) uses the following covariance matrices:

$$\check{P}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1.25\text{e}{-6} & 0 & 0\\ 0 & 0 & 0 & 0 & 1.25\text{e}{-6} & 0\\ 0 & 0 & 0 & 0 & 0 & 1.25\text{e}{-6}\end{bmatrix} \tag{A.12}$$

$$Q = \begin{bmatrix} 5\text{e}{-6} & 0 & 0 & 0 & 0 & 0\\ 0 & 5\text{e}{-6} & 0 & 0 & 0 & 0\\ 0 & 0 & 5\text{e}{-6} & 0 & 0 & 0\\ 0 & 0 & 0 & 3\text{e}{-8} & 0 & 0\\ 0 & 0 & 0 & 0 & 3\text{e}{-8} & 0\\ 0 & 0 & 0 & 0 & 0 & 3\text{e}{-8}\end{bmatrix} \tag{A.13}$$

$$R = \begin{bmatrix} 1\text{e}{-4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1\text{e}{-4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1\text{e}{-4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 3\text{e}{-2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 2\text{e}{-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 2\text{e}{-1} \end{bmatrix} \tag{A.14}$$

The derivation of the Λ matrix from Chapter 5 is given on the following page.

$$\Lambda \equiv \Xi(\hat{\bar{q}})^T \boldsymbol{\mathcal{L}}(\bar{q}(\hat{\boldsymbol{\omega}}_k))\Xi(\hat{\bar{q}}) = \begin{bmatrix} q_4 & q_3 & -q_2 & -q_1 \\ -q_3 & q_4 & q_1 & -q_2 \\ q_2 & -q_1 & q_4 & -q_3 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2}\hat{\omega}_3 & -\frac{1}{2}\hat{\omega}_2 & \frac{1}{2}\hat{\omega}_1 \\ -\frac{1}{2}\hat{\omega}_3 & 1 & \frac{1}{2}\hat{\omega}_1 & \frac{1}{2}\hat{\omega}_2 \\ \frac{1}{2}\hat{\omega}_2 & -\frac{1}{2}\hat{\omega}_1 & 1 & \frac{1}{2}\hat{\omega}_3 \\ -\frac{1}{2}\hat{\omega}_1 & -\frac{1}{2}\hat{\omega}_2 & -\frac{1}{2}\hat{\omega}_3 & 1 \end{bmatrix} \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \tag{A.15}$$

$$= \begin{bmatrix} q_4 + \frac{1}{2}q_1\hat{\omega}_1 - \frac{1}{2}q_2\hat{\omega}_2 - \frac{1}{2}q_3\hat{\omega}_3 & \frac{1}{2}q_3 + \frac{1}{2}q_1\hat{\omega}_2 + \frac{1}{2}q_2\hat{\omega}_1 + \frac{1}{2}q_4\hat{\omega}_3 & \frac{1}{2}q_1\hat{\omega}_3 - \frac{1}{2}q_2 + \frac{1}{2}q_3\hat{\omega}_1 - \frac{1}{2}q_4\hat{\omega}_2 & \frac{1}{2}q_3\hat{\omega}_2 - \frac{1}{2}q_2\hat{\omega}_3 - q_1 + \frac{1}{2}q_4\hat{\omega}_1 \\ \frac{1}{2}q_1\hat{\omega}_2 - q_3 + \frac{1}{2}q_2\hat{\omega}_1 - \frac{1}{2}q_4\hat{\omega}_3 & q_4 - \frac{1}{2}q_1\hat{\omega}_1 + \frac{1}{2}q_2\hat{\omega}_2 - \frac{1}{2}q_3\hat{\omega}_3 & q_1 + \frac{1}{2}q_2\hat{\omega}_3 + \frac{1}{2}q_3\hat{\omega}_2 + \frac{1}{2}q_4\hat{\omega}_1 & \frac{1}{2}q_1\hat{\omega}_3 - q_2 - \frac{1}{2}q_3\hat{\omega}_1 + \frac{1}{2}q_4\hat{\omega}_2 \\ q_2 + \frac{1}{2}q_1\hat{\omega}_3 + \frac{1}{2}q_3\hat{\omega}_1 + \frac{1}{2}q_4\hat{\omega}_2 & \frac{1}{2}q_2\hat{\omega}_3 - q_1 + \frac{1}{2}q_3\hat{\omega}_2 - \frac{1}{2}q_4\hat{\omega}_1 & q_4 - \frac{1}{2}q_1\hat{\omega}_1 - \frac{1}{2}q_2\hat{\omega}_2 + \frac{1}{2}q_3\hat{\omega}_3 & \frac{1}{2}q_2\hat{\omega}_1 - \frac{1}{2}q_1\hat{\omega}_2 - q_3 + \frac{1}{2}q_4\hat{\omega}_3 \end{bmatrix}$$

$$\times \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \tag{A.16}$$

$$= \begin{bmatrix} q_1^2 + q_2^2 + q_3^2 + q_4^2 & \frac{1}{2}\hat{\omega}_3(q_1^2 + q_2^2 + q_3^2 + q_4^2) & -\frac{1}{2}\hat{\omega}_2(q_1^2 + q_2^2 + q_3^2 + q_4^2) \\ -\frac{1}{2}\hat{\omega}_3(q_1^2 + q_2^2 + q_3^2 + q_4^2) & q_1^2 + q_2^2 + q_3^2 + q_4^2 & \frac{1}{2}\hat{\omega}_1(q_1^2 + q_2^2 + q_3^2 + q_4^2) \\ \frac{1}{2}\hat{\omega}_2(q_1^2 + q_2^2 + q_3^2 + q_4^2) & -\frac{1}{2}\hat{\omega}_1(q_1^2 + q_2^2 + q_3^2 + q_4^2) & q_1^2 + q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \tag{A.17}$$

$$= \begin{bmatrix} 1 & \frac{1}{2}\hat{\omega}_3 & -\frac{1}{2}\hat{\omega}_2 \\ -\frac{1}{2}\hat{\omega}_3 & 1 & \frac{1}{2}\hat{\omega}_1 \\ \frac{1}{2}\hat{\omega}_2 & -\frac{1}{2}\hat{\omega}_1 & 1 \end{bmatrix} \tag{A.18}$$

Bibliography

[1] Tewari, A., "Atmospheric and Space Flight Dynamics," Birkhäuser, 2007.

[2] Björck, A., "Numerical Methods for least squares problems," SIAM, Philadelphia. 1996.

[3] Dam, E.B, Koch, M., Lillholm, M. "Quaternions, Interpolation and Animation" Technical Report DIKU-TR-98/5, July 17, 1998. University of Copenhagen.

[4] Rogers, R.M., *Applied Mathematics in Integrated Navigation Systems, Third Edition.* Reston, VA: American Institute of Aeronautics and Astronautics, Inc., 2007.

[5] Kalman, R.E. "A New Approach to Linear Filtering and Prediction Problems" Transactions of the ASME–Journal of Basic Engineering, 1960. Vol 82, No Series D, pp. 35–45.

[6] Lefferts, E.J., Markley, F.L., and Shuster, M.D., "Kalman Filtering for Spacecraft Attitude Estimation," Paper 82-0070 presented at the AIAA 20th Aerospace Sciences Meeting, Jan 11-14, 1982, (also *AIAA Progress in Astronautics and Aeronautics* Vol 5, No 5, Sept-Oct 1982, pp. 417-429).

[7] Breckenridge, W.G., "Quaternions - Proposed Standard Conventions," JPL, INTEROFFICE MEMORANDUM IOM 343-79-1199, 1999.

[8] Trawny, N., and Roumeliotis, S.I., "Indirect Kalman Filter for 3D Attitude Estimation," MARS LAB Technical Report, Number 2005-002, Rev. 57 Dept. of Computer Science & Engineering, University of Minnesota

[9] Shimkin, N. "Estimation and Identification in Dynamical Systems," Lecture Notes, Fall 2009. Technion - Israel Institute of Technology.