Precision Navigation for Aerospace Applications

by

Andrew K. Stimac

B.S. Mechanical Engineering
Massachusetts Institute of Technology, 1999

SUBMITTED TO THE DEPARMENT OF MECHANICAL ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN MECHANICAL ENGINEERING
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2004

This work sponsored by the Department of the Air Force under Air Force contract #F19628-00-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the author and are not necessarily endorsed by the United States Government.

Signature of Author: _____
Department of Mechanical Engineering
May 7, 2004

Certified by: _____
David L. Trumper
Associate Professor of Mechanical Engineering
Thesis Supervisor

Accepted by: _____
Ain A. Sonin
Chairman, Department Committee on Graduate Students

Precision Navigation for Aerospace Applications

by

Andrew K. Stimac

**Submitted to the Department of Mechanical Engineering
on May 7, 2004 in Partial Fulfillment of the
Requirements for the Degree of Master of Science in
Mechanical Engineering**

Abstract

Navigation is important in a variety of aerospace applications, and commonly uses a blend of GPS and inertial sensors. In this thesis, a navigation system is designed, developed, and tested. Several alternatives are discussed, but the ultimate design is a loosely-coupled Extended Kalman Filter using rigid body dynamics as the process with a small angle linearization of quaternions. Simulations are run using real flight data. A bench top hardware prototype is tested. Results show good performance and give a variety of insights into the design of navigation systems. Special attention is given to convergence and the validity of linearization.

Thesis Supervisor: David L. Trumper
Title: Associate Professor of Mechanical Engineering

Biographical Note

Andrew Stimac is from Seattle, Washington and holds a Bachelor's Degree in Mechanical Engineering from MIT in 1999. This thesis was conducted in conjunction with MIT Lincoln Laboratory, and is in partial fulfillment for a Master's Degree in Mechanical Engineering at MIT in 2004.

Andrew's research interest is the intelligent control of mechanical systems. This includes modern control theory with particular attention to nonlinear systems and adaptive techniques.

Andrew's recreational activities include sand volleyball, alpine skiing, and music. He now lives in Acton, Massachusetts with his wife Amy.

Acknowledgements

I give special recognition to those who have inspired, educated, and assisted me in this project.

Above all, I would like to thank my advisors Prof. David Trumper and Anthony Hotz for their guidance. I am also deeply indebted to Deborah Blanchard and Leonas Bernotas at MIT Lincoln Laboratory for the excellent engineering that made this project possible. Finally, I am grateful to my MIT instructors Prof. Harry Asada and Prof. Jean-Jacques Slotine for their enlightening illustration of the subject matter.

Table of Contents

List of Figures

List of Tables

1. Introduction

The problem of navigation is a fundamental engineering challenge. The objective is to estimate the position and attitude of a moving vehicle using the available measurements. This is primarily a task in discriminating between information and noise in a dynamic system.

The field of navigation is both mature and evolving. Successful space programs, including the Apollo missions to the moon, are the result of major innovations in this area. Today, modern navigation systems have been applied to a broad range of vehicles for strategic and recreational purposes. Still, this technology must continue to progress in order to meet the demands of future scientific, military, and commercial applications.

At MIT Lincoln Laboratory, there are several current programs that motivate research in this area. Airborne sensors, such as laser radar, require a high degree of navigation accuracy for the interpretation and processing of their images. Other applications include laser communications and aerospace guidance systems. All of these projects have the common requirement of precision pointing and positioning.

In the design of this particular navigation system, there are several goals and areas of focus. The first is to understand and improve the treatment of attitude, which is increasingly important in modern applications. Large angle motion requires nonlinear representation, such as Euler angles or quaternions. Attitude also experiences varying degrees of observability in a GPS-aided inertial system. For these reasons, the representation of attitude warrants special attention so that accuracy can be improved and the causes of error can be determined.

The second task is to develop tighter integration between the navigation system and the onboard controller. It is typical for the navigation system to be a distinct entity that functions autonomously to provide a navigation solution for the controller. However, the controller contains actuator commands and other information about the vehicle dynamics. The proposal here is to use these control signals in the navigation solution. This has potential benefits for improving the dynamic response.

The third goal is to achieve increased flexibility. Off-the-shelf navigation systems are usually closed boxes with proprietary algorithms, which makes them difficult to integrate and tune. This project creates source code that can be modified to meet a great variety of applications. It can also serve as a tool for analyzing and comparing off-the-shelf systems.

Underlying these efforts, a number of existing techniques are explored and evaluated. The Kalman filter is used as the predominant method of estimation, because it is well suited and time-tested for this particular problem. While several assumptions must be satisfied concerning linearity and white noise, the result is on firm theoretical ground and potentially outperforms most practical alternatives.

The scope of this project is limited to the integration of an Inertial Measurement Unit (IMU) with a Global Position System (GPS) receiver. These are the current leading technologies

and the most frequently used. However, the results here can be extended to include additional sensors. In fact, discussion in this report motivates the use of additional measurements.

1.1 Summary of Procedure

This project addresses navigation systems from three distinct vantage points: theory, simulation, and hardware. Each provides its own benefits and limitations. The combination of the three produces a strong, balanced analysis. The thesis is organized as follows:

The theoretical derivation is presented in Section 2. This begins with background on navigation and establishes the fundamental navigation equations. Subsequently, the Kalman filter is discussed and derived. The derivation is specific to the form used in this project, and explores several relevant details. The topics of linearization, data forgetting, and convergence are discussed. Finally, the application of the Kalman filter to navigation is presented. This details the decisions of modeling and design, and the calculation of the state matrices and propagation equations. The observability of the system is analyzed.

Simulation of the navigation system is conducted in order to characterize behavior. A variety of studies and results are presented in Section 3. Performance is evaluated using flight datasets from an aircraft and a booster rocket. Analysis is also performed using simulated data, and a large range of operating conditions and parameters is explored. Accuracy is accessed using several different methods including the filter covariance, comparison to a reference solution, and comparison to the inertial solution. Filter health and stability are explored.

Figure 1-1: Benchtop Testing Environment

Figure 1-2: Hardware Components

A hardware prototype is constructed in order to evaluate this design on an embedded real-time system. The benchtop setup is shown in Figure 1-1 and the components are shown in Figure 1-2. The filter algorithm is ported onto a high-performance Digital Signal Processor made by Texas Instruments (SMJ320C6701). Computational efficiency is investigated and dramatically improved in order to achieve an acceptable sample rate. This activity is presented in Section 4.

The collective results of these activities are discussed in Section 5. Results from theory, simulation, and hardware are combined to evaluate the design decisions of the filter. Opinions are presented regarding the success and shortcomings of these techniques.

Recommendations for immediate and future work are given in Section 6. This begins with a summary of the current capabilities and suggestions for their utilization. Then several methods of improvement are discussed for both short-term and long-term development. These recommendations serve as a guide for future research in this area.

Finally, Section 7 gives a brief conclusion.

1.2 Summary of Key Results

This section describes some of the key results of this thesis.

The Extended Kalman Filter (EKF), which uses a linearization about the estimated state, has been chosen for this application. This approach is justified theoretically, and the results of simulation and hardware confirm that convergence and stability are achieved. The EKF is well suited to this application and provides an accurate solution. Exponential data forgetting improves filter response in many anomalous conditions.

The attitude representation for this filter uses a combination of quaternions and small angle errors. The theoretical derivation shows that small angles can provide a concise linearization for the Kalman filter, while the quaternions give singularity-free attitude propagation. This leads to the minimal three attitude variables in the Kalman filter. With careful implementation, the use of small angles does not reduce the accuracy of the Kalman filter. This is because the EKF already requires linearization and the small angles errors are immediately offloaded to the quaternions.

Rigid body dynamics are used as the process model. With this architecture, force and torque are inputs to the model. The model states include position, orientation, and their derivatives. Newton's laws of motion for rigid bodies are used to propagate the state.

This technique can improve accuracy in cases where the dynamics are predictable. Direct force and torque control inputs are provided so that known values can be given to the filter. Unknown force and torque disturbances are estimated by the filter. First order noise processes are used to represent the time correlation in the disturbance noise.

Unfortunately, this approach greatly complicates the system model and produces a significant computational burden for real-time hardware implementation. For applications with unpredictable dynamics, it is preferable to use an inertial navigation solution as the process for the Kalman filter.

Conditions for the convergence of the state estimate are established using observability analysis. Several interesting results are derived theoretically and confirmed in simulation. They can be summarized as follows:

- Body rotation rate, position, velocity, and force and torque disturbances are all observable using IMU and GPS measurements.
- Attitude rotation is observable about the direction of applied force and becomes completely observable when the force is varied with respect to an inertial coordinate frame. This applied force does not include gravity and is in the same direction as the measured acceleration.
- IMU biases in the gyroscopes and accelerometers are completely observable when the force is varied with respect to the body coordinate frame. This may be done under constant force by rotating the vehicle or the IMU.
- Gyroscope bias observability is improved with knowledge that the vehicle is Earth stationary or by the precession of a rotating rigid body.
- A gravity model bias can be distinguished from an accelerometer bias if there is sufficient body rotation.

Covariance analysis provides the expected filter accuracy using the noise values of the actual process and measurements. In this method, the error covariance matrix of the Kalman filter is propagated numerically. Accuracy is dependent on the flight path complexity, in agreement with the observability results outlined above. The sensitivity of the estimate accuracy to measurement noise and other parameters is established. This provides conclusive predictions that are useful for sensor selection and system design.

The navigation algorithm has been successfully implemented on real-time embedded hardware. However, difficulties were encountered during this process and several implementation issues have been uncovered.

Great improvements in computation and memory efficiency were required for implementation. They were achieved using a variety of techniques for an ultimate sample rate of 100 Hz. However, several sacrifices were made that ultimately caused a decrease in accuracy and robustness. Specific improvements to the algorithm and model are recommended for future hardware implementation.

We now begin the theoretical derivation of this thesis.

2. Theoretical Derivation

The theory presented in this section contains two primary topics: navigation, and the Kalman filter. These topics are treated separately, in Sections 2.1 and 2.2 respectively, and then considered jointly in Section 2.3.

The discussion of navigation in Sections 2.1 establishes the background theory. Coordinate systems for representing position and attitude are established and equations for vehicle dynamics are developed. Modeling of the Earth's shape, rotation, and gravitational field is discussed. Techniques for initial alignment and for computing a pure inertial solution are given.

A full derivation of the Kalman filter is given Section 2.2, with special attention to the details relevant to this application. This includes the topics of data forgetting and application to nonlinear systems. A continuous-time process with discrete-time measurements is used. Methods of analysis and implementation are also presented.

Section 2.3 describes the application of the Kalman filter to the navigation problem. Modeling and the selection of state variables are discussed. Jacobians are calculated in closed form and the system is linearized. The observability of the system is analyzed theoretically.

2.1 Navigation

This discussion begins with the underlying theory of navigation. The material in this section summarizes the geometry and physics that pertain to an airborne or terrestrial vehicle in motion. These equations are central to the function and analysis of this navigation system.

2.1.1 Coordinate Systems

A series of coordinate systems can be developed to represent vehicle motion. This development is intended for navigation with respect to the Earth, and as a result a model of the Earth will be required. Cartesian frames will be of primary importance, but spherical and elliptical coordinates will be useful for representing points on the Earth.

Earth Centered Inertial (ECI)

An inertial coordinate system is required to describe the dynamics of motion, and ECI serves this purpose. The ECI frame has its origin at the center of the Earth, but does not rotate with the Earth. The combined effect of ignoring both the Sun's gravity and the Earth's orbital motion is negligible. Written as a position vector

$$p_{eci} = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^T, \tag{2-1}$$

the direction of p_z is upward through the North Pole. The directions of p_x and p_y lie within the Earth's equatorial plane, forming a right handed frame. The direction of p_x is most conveniently defined with respect to a longitude on the Earth at a particular time. For example, we could use the launch site longitude at launch time or the Greenwich meridian during the vernal equinox.

Earth Centered Fixed (ECF)

ECF are Cartesian coordinates that rotate with the Earth. As such, they are no longer inertial, but are useful for defining motion relative to the Earth. The relation between ECF and ECI can be expressed using a rotation matrix R_e:

$$p_{ecf} = R_e p_{eci}, \tag{2-2}$$

given by

$$R_e = \begin{bmatrix} \cos\theta_e & \sin\theta_e & 0 \\ -\sin\theta_e & \cos\theta_e & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2-3}$$

where θ_e is the Earth angle. The Earth angle propagates at Earth rate [14]

$$|\omega_e| = 7.2921150\times10^{-5} \text{ rad/sec}, \tag{2-4}$$

in the direction of right-handed rotation about the positive *z*-axis in both ECI and ECF. Earth north nutation is ignored in most applications. An initial condition for θ_e can determine the relative alignment of the ECI and ECF frames.

Geodetic and Geocentric

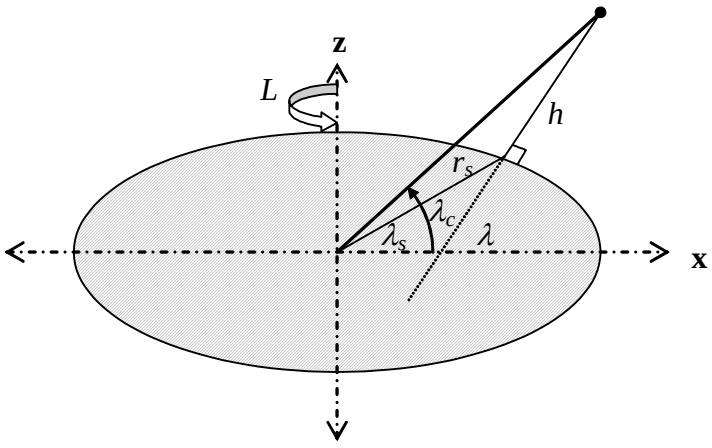

Figure 2-1: Geodetic and Geocentric Coordinates [20]

The shape of the Earth is well approximated by an ellipsoid, which is the foundation of geodetic coordinates. However, simpler mathematical expressions result from a spherical model, which is used by geocentric coordinates. Both coordinates are shown in Figure 2-1. The position vector *p*, in either ECI or ECF, can be represented by its geocentric latitude λ_c and radius *r*, or by its geodetic latitude λ and altitude *h* (from the ellipsoid). The geodetic ellipsoid is usually defined by the World Geodetic Survey [14], and has major axis $r_e = 6378137.0$ meters along the equator and minor axis $(1-\mu)r_e$ at the poles, with the flattening factor $\mu = 1/298.257223563$.

The conversion from Cartesian to geocentric coordinates is a spherical transform:

$$r = \sqrt{p_x{}^2 + p_y{}^2 + p_z{}^2} \tag{2-5}$$

$$\lambda_c = \tan^{-1}\left(\frac{p_z}{\sqrt{p_x^2 + p_y^2}}\right) = \text{atan2}\left(p_z, \sqrt{p_x^2 + p_y^2}\right) \tag{2-6}$$

$$L = \tan^{-1}\left(\frac{p_y}{p_x}\right) = \text{atan2}\left(p_y, p_x\right), \tag{2-7}$$

where the four quadrant arctangent is used to eliminate singularities. The longitude *L* will be inertial (celestial) if ECI position is used, and fixed to the Earth if ECF position is used.

The inverse transform is

$$p_x = r\cos(L)\sin(\lambda_c) \tag{2-8}$$

$$p_y = r\sin(L)\sin(\lambda_c) \tag{2-9}$$

$$p_z = r\cos(\lambda_c) \tag{2-10}$$

The conversion between geocentric and geodetic coordinates is somewhat more involved. Geodetic and geocentric longitude are identical. Returning to Figure 2-1, the projection of *p* onto this ellipsoid has geocentric coordinates λ_s and r_s, where the subscript stands for "sea-level". These quantities can be related to the geodetic coordinates using trigonometry and the equation of the ellipsoid:

$$\lambda_s = \tan^{-1}\left((1-\mu)^2 \tan\lambda\right) = \text{atan}\,2\left((1-\mu)^2 \sin\lambda, \cos\lambda\right) \tag{2-11}$$

$$r_s = \frac{r_e}{\sqrt{1+\left((1-\mu)^{-2}-1\right)\sin^2\lambda_s}}. \tag{2-12}$$

Using vector addition of the sea-level radius and altitude, the result is

$$r = \sqrt{r_s^{\,2} + h^2 + 2r_s h\cos(\lambda_s - \lambda)} \tag{2-13}$$

$$\lambda_c = \tan^{-1}\left(\frac{r_s \sin\lambda_s + h\sin\lambda}{r_s\cos\lambda_s + h\cos\lambda}\right). \tag{2-14}$$

The inverse transform requires solving a quartic, but can be expressed in closed form [18, 19]. This result is somewhat cumbersome and a preferable alternative is the iterative solution

$$\hat{h}_{k+1} = \hat{h}_k - \left(\hat{r}_k - r_k\right) \tag{2-15}$$

$$\hat{\lambda}_{k+1} = \hat{\lambda}_k - \left(\hat{\lambda}_{ck} - \lambda_{ck}\right). \tag{2-16}$$

Here $\hat{h}_k$ and $\hat{\lambda}_k$ are the estimated geodetic altitude and latitude, respectively, at sample *k*, and $\hat{r}_k$ and $\hat{\lambda}_{ck}$ are the geocentric estimates calculated using (2-13) and (2-14). The geocentric estimation errors are used to drive the geodetic estimate. It can be shown that convergence is exact for stationary vehicles and reasonably accurate when the iterations are fast compared to vehicle motion. This approach is successful because the Earth is almost spherical.

North East Down (NED)

NED is a local level Cartesian frame defined by Geodetic coordinates. North is the direction of increasing latitude, east is the direction of increasing longitude, and down is the direction of decreasing altitude. The NED origin is typically unimportant, but can be defined as the projection of the vehicle position onto the ellipsoid. As such, the NED origin is dependent on the current position, so NED position is not meaningful. However, the NED frame provides an intuitive description of velocity and attitude.

The combination of gravity and centrifugal force that an object feels when stationary on the Earth's surface is very closely aligned to the NED z-axis (down). This is just a consequence of the Earth's surface being in equilibrium. If the gravity direction were not downwards, the mass at the Earth's surface would slowly redistribute.

East North Up (ENU) is a similar alternative. However, NED has preferred properties related to attitude representation, which are discussed later.

Body

The body frame has axes fixed to the vehicle with origin at the vehicle center of mass. As such, the body frame rotates with the vehicle and is meaningful for describing the vehicle inertia and the locations of thrusters, sensors, and other components. The orientation of the body frame represents the vehicle orientation.

The typical choice of axes for aerospace applications is *x* forward (axial), *y* starboard, and *z* down. Body coordinates are frequently used to describe body rotation, with the names roll, pitch, and yaw given to rotations about the body *x*, *y*, and *z* axes, respectively.

Body coordinates are aligned with NED if an aircraft is level and facing north. Here pitch and yaw match the common definitions of elevation and azimuth.

2.1.2 Attitude Representation

Attitude representation describes the orientation of one Cartesian frame with respect to a second reference frame. In this application, attitude represents the orientation of the vehicle body frame, which can be referenced to the ECI, ECF, or NED frame. However, this discussion also provides methods for converting between any two coordinate systems, such as from ECI to NED. Much of the discussion will use the terms body and reference frame, but can be applied to any set of frames.

We will explore four alternative methods: Euler angles, rotation vectors, quaternions, and small angles. For each of these, the Direction Cosine Matrix (DCM), denoted *B*, can be calculated. This matrix gives the rotation from the reference frame to the body frame, so that we can write

$$x_b = Bx\,, \tag{2-17}$$

where *x* is a 3 dimensional vector in the reference frame and x_b is a 3 dimensional vector in the body frame. As a rotation matrix, *B* is orthogonal so that $B^{-1} = B^T$ and therefore

$$x = B^T x_b\,. \tag{2-18}$$

The translation between coordinate systems can be treated separately, but is typically less important for this application. This is because the origin is irrelevant when discussing velocity or orientation, and most of our position representations share a common origin at the center of the Earth.

It will also be required to derive an equation for attitude propagation under vehicle rotation. The rate of rotation of the body frame is usually represented, and measured, in body coordinates. A general relation is developed below, which will later be applied specifically to each attitude representation.

Consider a vector x_b that is fixed in body coordinates. The inertial derivative, meaning the rate of change with respect to the reference frame but expressed in body frame coordinates, is given by

$$\left(\frac{dx_b}{dt}\right)_{inertial} = \dot{x}_b + \omega \times x_b\,. \tag{2-19}$$

This is the sum of motion relative to the body frame, which is zero, and motion caused by body frame rotation. Transforming this result back to the reference frame gives

$$\dot{x} = B^T(\omega \times x_b) = B^T[\omega\times]x_b, \tag{2-20}$$

where the vector cross product can be written in matrix form using

$$[\omega\times] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \tag{2-21}$$

This matrix expansion of the vector cross product is convenient throughout the analysis. Another more direct differentiation is

$$\dot{x} = \frac{d}{dt}\left(B^T x_b\right) = \dot{B}^T x_b + B^T \dot{x}_b = \dot{B}^T x_b. \tag{2-22}$$

Equating (2-20) and (2-22) yields

$$\dot{B}^T x_b = B^T[\omega\times]x_b. \tag{2-23}$$

For this to hold for all x_b, we must have

$$\dot{B}^T = B^T[\omega\times], \tag{2-24}$$

Since the cross product is skew-symmetric, this is equivalent to

$$\dot{B} = -[\omega\times]B. \tag{2-25}$$

Using (2-25), orientation can be propagated using the entire 3 by 3 *B* matrix, although this is highly redundant because 9 terms are being used to represent 3 degrees of freedom. Still, this result will be useful for developing propagation equations for specific attitude representations.

Euler Angles

A logical approach is to represent orientation as a sequence of large-angle, single-axis, body rotations. Define an Euler vector as $\gamma = [\varphi \quad \theta \quad \psi]^T$, containing roll, pitch, and yaw, respectively, as shown in Figure 2-2. These rotations are right-handed about the body *x*, *y*, and *z* axes, respectively, which is why $-\theta$ is shown in the figure.

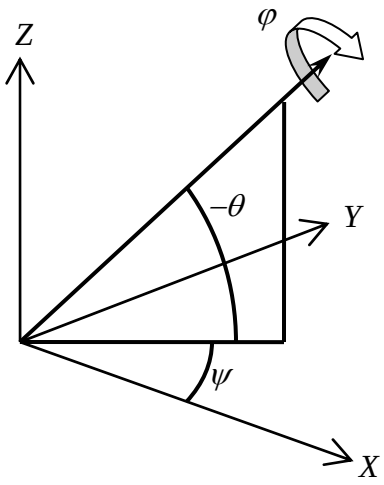

Figure 2-2: Euler Angle Representation

Because large rotations do not commute, a convention for the order of rotation is required. The most common order is yaw, pitch, roll. Note that the new body axes are used for each successive rotation. The DCM can be expressed as the product of three rotation matrices:

$$B(\gamma)=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\varphi & \sin\varphi\\ 0 & -\sin\varphi & \cos\varphi\end{bmatrix}\begin{bmatrix}\cos\theta & 0 & -\sin\theta\\ 0 & 1 & 0\\ \sin\theta & 0 & \cos\theta\end{bmatrix}\begin{bmatrix}\cos\psi & \sin\psi & 0\\ -\sin\psi & \cos\psi & 0\\ 0 & 0 & 1\end{bmatrix} \tag{2-26}$$

Performing the matrix multiplication yields [20, p. 37]

$$B(\gamma)=\begin{bmatrix}\cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta\\ \sin\varphi\sin\theta\cos\psi-\cos\varphi\sin\psi & \sin\varphi\sin\theta\sin\psi+\cos\varphi\cos\psi & \sin\varphi\cos\theta\\ \cos\varphi\sin\theta\cos\psi+\sin\varphi\sin\psi & \cos\varphi\sin\theta\sin\psi-\sin\varphi\cos\psi & \cos\varphi\cos\theta\end{bmatrix}. \tag{2-27}$$

Inspection of this matrix reveals a method for extracting the Euler angles from the components B_{ij} of the DCM. Consider using the simple terms on the top row and right column:

$$\varphi=\tan^{-1}\left(B_{23}/B_{33}\right)=\text{atan2}\left(B_{23},B_{33}\right) \tag{2-28}$$

$$\theta=-\sin^{-1}B_{13} \tag{2-29}$$

$$\psi=\tan^{-1}\left(B_{12}/B_{11}\right)=\text{atan2}\left(B_{12},B_{11}\right). \tag{2-30}$$

These equations can also serve as the conversion to Euler angles if the DCM was calculated using other means, as are discussed later. All that is required is the DCM is, in fact, a rotation matrix.

Now we calculate the Euler update under body rotation using the relation

$$\dot{B}(\gamma) = -[\omega\times]B(\gamma). \tag{2-31}$$

The right side can be evaluated using our earlier expression for $B(\gamma)$. The multiplication is somewhat involved, but only the following entries will be required

$$\dot{B}_{11} = \omega_3(\sin\varphi\sin\theta\cos\psi - \cos\varphi\sin\psi) - \omega_2(\cos\varphi\sin\theta\cos\psi + \sin\varphi\sin\psi) \tag{2-32}$$

$$\dot{B}_{12} = \omega_3(\sin\varphi\sin\theta\sin\psi - \cos\varphi\cos\psi) - \omega_2(\cos\varphi\sin\theta\sin\psi + \sin\varphi\cos\psi) \tag{2-33}$$

$$\dot{B}_{13} = \omega_3\sin\varphi\cos\theta - \omega_2\cos\varphi\cos\theta \tag{2-34}$$

$$\dot{B}_{23} = \omega_3\sin\theta + \omega_1\cos\varphi\cos\theta \tag{2-35}$$

$$\dot{B}_{33} = \omega_2\sin\theta - \omega_1\sin\varphi\cos\theta \tag{2-36}$$

The Euler extraction equations (2-28 to 2-30) can also be differentiated:

$$\dot{\varphi} = \frac{d}{dt}\tan^{-1}\left(\frac{B_{23}}{B_{33}}\right) = \frac{B_{33}\dot{B}_{23} - B_{23}\dot{B}_{33}}{B_{33}^2 + B_{23}B_{33}} \tag{2-37}$$

$$\dot{\theta} = \frac{d}{dt}\left(-\sin^{-1}B_{13}\right) = \frac{-\dot{B}_{13}}{\sqrt{1-B_{13}^2}} \tag{2-38}$$

$$\dot{\psi} = \frac{d}{dt}\tan^{-1}\left(\frac{B_{12}}{B_{11}}\right) = \frac{B_{11}\dot{B}_{12} - B_{12}\dot{B}_{11}}{B_{11}^2 + B_{12}B_{11}} \tag{2-39}$$

Substitution of (2-32) through (2-36) into (2-37), (2-38), and (2-39) leads to substantial reduction and can be expressed in matrix form

$$\dot{\gamma} = \begin{bmatrix} 1 & \tan\theta\sin\varphi & \tan\theta\cos\varphi \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi\sec\theta & \cos\varphi\sec\theta \end{bmatrix}\omega. \tag{2-40}$$

This update is in agreement with spatial intuition. Rotation in ω_1 only effects roll, because roll is the third rotation in the Euler angles. The direction of pitch is a combination of ω_2 and ω_3, depending on the roll angle. Yaw has a similar dependency but the sine and cosine are reversed, so that the effect of roll is offset by 90°. Also, the term $\sec\theta$ causes yaw motion to diminish at large pitch angles. This is analogous to the meridians of the Earth becoming close together near the poles. The additional terms linking ω_2 and ω_3 to roll are the result of roll-yaw coupling as the vehicle moves around the pole and are equivalent to $\dot{\psi}\sin\theta$.

Note that the roll and yaw updates are discontinuous when $\theta = \pm 90^\circ$. This singularity makes Euler angles unsuitable for propagation in this vicinity. However, they are perhaps the most intuitive description of attitude, and remain useful for input and output. Alternatives are explored below.

Rotation Vectors

It is a somewhat surprising fact, attributed to Euler, that any orientation can be expressed as a single rotation about a particular axis. A geometric method for constructing this axis, and thereby proving this theorem, is given in [23]. This rotation can be described using a 3-dimensional rotation vector

$$\rho = \begin{bmatrix} \rho_1 & \rho_2 & \rho_3 \end{bmatrix}^T , \tag{2-41}$$

where the magnitude $|\rho|$ is the angle of rotation and the vector direction is the axis of rotation. This technique is explored in detail in [8, pp. 347-361], although the definition of ρ has the sign reversed.

The DCM can be calculated from the matrix differential equation (2-25)

$$\dot{B} = -[\omega \times] B . \tag{2-42}$$

Consider a motion of gradual increase in rotation

$$\rho(t) = \rho_o t , \tag{2-43}$$

so that $\rho(t) = \rho_o$ at $t = 1$. Since the rotation vector remains constant in both frames, the body rates are constant and are given by

$$\omega = \frac{d\rho}{dt} = \rho_o . \tag{2-44}$$

Therefore, the differential equation of (2-42) becomes

$$\dot{B} = -[\rho_o \times] B \tag{2-45}$$

which has a solution using the exponential of the cross product matrix:

$$B(t) = C \exp\left(-[\rho_o \times] t\right). \tag{2-46}$$

The initial condition of $B(0) = I$ leads to $C = I$, and the value at $t = 1$ is then

$$B(1) = B(\rho_o) = \exp[-\rho_o \times]. \tag{2-47}$$

In general, for any ρ, the DCM matrix is given by the matrix exponential

$$B(\rho) = \exp[-\rho \times]. \tag{2-48}$$

For direct evaluation, this expression can be rewritten using the infinite series expansion of the exponential, sine, and cosine [8, p. 348]:

$$B(\rho) = \cos|\rho| I + \frac{1-\cos|\rho|}{\rho^T \rho}\rho\rho^T + \frac{\sin|\rho|}{|\rho|}[-\rho\times] \tag{2-49}$$

Note that for small rotations this expression approaches

$$B(\rho) \approx I - [\rho\times]. \tag{2-50}$$

Extraction equations can also be formulated. The propagation equations in terms of ρ are very cumbersome, although they appear to be singularity free. The decision here is to avoid this representation. The complexity in the underlying equations is unsuitable for programming and complicates implementation.

Quaternions

The complexities of the rotation vector representation can be avaoided by adding a fourth parameter. This can be done using the magnitude $|\rho|$ and unit vector $\rho/|\rho|$. However, the ultimate result is simplified using the following substitutions [20, p. 41]:

$$q_0 = \cos(\rho/2) \tag{2-51}$$
$$q_1 = (\rho_1/|\rho|)\sin(\rho/2) \tag{2-52}$$
$$q_2 = (\rho_2/|\rho|)\sin(\rho/2) \tag{2-53}$$
$$q_3 = (\rho_3/|\rho|)\sin(\rho/2), \tag{2-54}$$

which are known as the quaternion parameters. Note that

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = \cos^2\left(\frac{\rho}{2}\right) + \left(\frac{\rho_1^2 + \rho_2^2 + \rho_3^2}{|\rho|^2}\right)\sin^2\left(\frac{\rho}{2}\right) = 1. \tag{2-55}$$

This illustrates that the four quaternions are not independent state variables, and also suggests a geometric analogy of a four dimensional sphere.

Calculation of the DCM using quaternions is conducted in full in [23] and summarized in [20, p. 41]. The result is now free of trigonometric functions, but still nonlinear

$$B(q) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}. \tag{2-56}$$

Conversion from quaternions to Euler angles is accomplished by forming $B(q)$, and then using the Euler extraction formula. It only necessary to calculate the top row and right column of $B(q)$.

The quaternions can be extracted from the DCM, but the sign of the solution is not unique. The elements on the main diagonal can be added and subtracted to form the following equations:

$$4q_0^2 = 1 + B_{11} + B_{22} + B_{33} \tag{2-57}$$
$$4q_1^2 = 1 + B_{11} - B_{22} - B_{33} \tag{2-58}$$
$$4q_2^2 = 1 - B_{11} + B_{22} - B_{33} \tag{2-59}$$
$$4q_3^2 = 1 - B_{11} - B_{22} + B_{33} \tag{2-60}$$

The off-diagonal elements provide information on the signs:

$$4q_0q_1 = B_{23} - B_{32} \tag{2-61}$$
$$4q_0q_2 = B_{31} - B_{13} \tag{2-62}$$
$$4q_0q_3 = B_{12} - B_{21} \tag{2-63}$$
$$4q_1q_2 = B_{12} - B_{21} \tag{2-64}$$
$$4q_2q_3 = B_{23} - B_{32} \tag{2-65}$$
$$4q_1q_3 = B_{13} - B_{31}\,. \tag{2-66}$$

Still, the first sign must be chosen arbitrarily. If the DCM was constructed from another attitude representation, extraction of the quaternions provides a conversion. For conversion from Euler angles to quaternions, this extraction can be simplified to the following equations [20, p. 41]:

$$q_0 = \pm\left(\cos(\varphi/2)\cos(\theta/2)\cos(\psi/2) + \sin(\varphi/2)\sin(\theta/2)\sin(\psi/2)\right) \tag{2-67}$$
$$q_1 = \pm\left(\sin(\varphi/2)\cos(\theta/2)\cos(\psi/2) - \cos(\varphi/2)\sin(\theta/2)\sin(\psi/2)\right) \tag{2-68}$$
$$q_2 = \pm\left(\cos(\varphi/2)\sin(\theta/2)\cos(\psi/2) + \sin(\varphi/2)\cos(\theta/2)\sin(\psi/2)\right) \tag{2-69}$$
$$q_3 = \pm\left(\cos(\varphi/2)\cos(\theta/2)\sin(\psi/2) - \sin(\varphi/2)\sin(\theta/2)\cos(\psi/2)\right), \tag{2-70}$$

where the same sign must be chosen for all four equations. This can be derived using the quaternion extraction equations and Euler DCM definition, but is most easily verified in reverse using trigonometric half-angle identities.

The propagation equation can be derived in a fashion similar to Euler propagation, using

$$\dot{B}(q) = -[\omega\times]B(q)\,. \tag{2-71}$$

The right side can be directly evaluated and compared to the derivative of the quaternion extraction equations. This result is bilinear in the state variables q and ω:

$$\dot{q} = -\frac{1}{2}\begin{bmatrix} q_1 & q_2 & q_3 \\ -q_0 & q_3 & -q_2 \\ -q_3 & -q_0 & q_1 \\ q_2 & -q_1 & -q_0 \end{bmatrix}\omega = f_q(q)\omega . \tag{2-72}$$

The magnitude constraint is consistent with this propagation. However, under imperfect numerical computation, it is usually important to renormalize the quaternions periodically.

The simplicity of the quaternion propagation equations and the absence of singularities make quaternions the preferred method of attitude representation. However, the use of four parameters, a non-minimal representation, is in fact a grave problem for observers and the Kalman filter. The four quaternions are not independent state variables and should not be treated as such. The normality constraint must be imposed in some fashion or the four parameter dynamics will be unobservable and incorrect.

Small Angle Rotations

Small angles provide a linear representation of small changes in orientation which are order-independent. They can be suitable for representing the attitude error, or similar small quantities. The small rotation is approximated using a 3-dimensional vector of small angles α in the direction of body rotations:

$$\delta\alpha = \omega\delta t . \tag{2-73}$$

Under the assumption of small angles, the DCM given for Euler angles (2-27) reduces to

$$R_\alpha(\alpha) = \begin{bmatrix} 1 & \alpha_3 & -\alpha_2 \\ -\alpha_3 & 1 & \alpha_1 \\ \alpha_2 & -\alpha_1 & 1 \end{bmatrix} = I - [\alpha\times]. \tag{2-74}$$

The inverse is approximately

$$R_\alpha^{-1} \approx R_\alpha^T = R_\alpha(-\alpha) = I + [\alpha\times] \tag{2-75}$$

Note that small angle rotations commute.

2.1.3 Gravity Model

The acceleration of Earth's gravity has been precisely modeled by several studies including the World Geodetic Survey of 1984 [14]. A spherical approximation is given by Newton's Law of Gravitation:

$$g = -GM_e \frac{p}{|p|^3}, \tag{2-76}$$

where $G = 6.67259 \times 10^{-11}$ m^3/kg-s^2 is a gravitational constant, and $M_e = 5.98 \times 10^{24}$ kg is the mass of the Earth. For linearization, the Jacobian will be needed, which can be expressed as

$$\frac{\partial g}{\partial p} = -\frac{GM_e}{|p|^3}\left(I - 3\frac{pp^T}{|p|^2}\right). \tag{2-77}$$

The Jacobian is a matrix of partial derivatives. In this 3 by 3 matrix, each row corresponds to a component of g while each column corresponds to a component of p. An elliptical model

$$g = -GM_e \frac{p}{\|p\|^3} - G_{hc}\frac{r_e^2}{\|p\|^2}\begin{pmatrix} 1-5\sin^2\lambda_c \\ 1-5\sin^2\lambda_c \\ 3-5\sin^2\lambda_c \end{pmatrix}, \tag{2-78}$$

provides greater accuracy, where G_{hc} is now the harmonic gravitational constant and λ_c is the geocentric latitude. A Jacobian for this model can also be calculated but it is usually acceptable to use the spherical Jacobian.

2.1.4 Rigid Body Dynamics

The physics of vehicle motion will be represented using rigid body dynamics. Aerodynamics forces can be represented using the force and torque input to this model. Additional modeling of vibration modes is not pursued here.

Rotation

Rotation is best analyzed in the body coordinate frame. Differentiating the angular momentum in a rotating frame gives

$$T = \frac{d}{dt}(J\omega) = J\dot{\omega} + \omega \times (J\omega), \tag{2-79}$$

where J is the inertia matrix. This leads to

$$\dot{\omega} = -J^{-1}(\omega \times (J\omega)) + \tau \tag{2-80}$$

where $\tau = J^{-1}T$ is the specific torque. Note that multiplication of J by a scalar has no effect on (2-80), except in the computation of specific torque.

Attitude is computed using one of the update laws given earlier, such as quaternions.

Translation

In ECI, translational dynamics follow directly from Newton's Law of Motion

$$\dot{p} = v \tag{2-81}$$

$$\dot{v} = B^T(q) f + g(p), \tag{2-82}$$

where f is the force given in body coordinates and must therefore be transformed using the inverse DCM. While it is possible to formulate these equations in other coordinate systems, such as ECF or geodetic, ECI has been chosen because of the simplicity of these equations. However, initialization and measurements that are relative to the Earth will now require more elaborate expressions.

2.1.5 Measurements

This section describes the primary measurements used in navigation. This includes an Inertial Measurement Unit (IMU) and the Global Positioning System (GPS). Addition measurements may also be available.

Inertial Measurement Unit (IMU)

This application uses a "strapdown" IMU, where the gyroscopes and accelerometers are fixed to the body frame. The gyroscope measurement vector ω_m represents the body rate ω in all 3 axes. The accelerometer measurement vector a_m represents the difference between vehicle acceleration and the acceleration of gravity, and is also expressed in body coordinates.

A pure inertial solution can be calculated directly from the IMU measurements. Substituting these measurements into the equations of motion, we have

$$\dot{q} = f_q(q)\omega_m \tag{2-83}$$

$$\dot{p} = v \tag{2-84}$$

$$\dot{v} = B^T(q)a_m + g(p), \tag{2-85}$$

which can be integrated in real time. Note that $\dot{\omega}$ is never calculated and is available only through differentiation. This solution will drift because errors accumulate in the integrators. The position/gravity relation is actually slightly unstable. An error in position will cause a gravity estimation error, which will misinterpret future accelerometer readings. As a result,

position estimation is poor from an INS. With high quality gyros, the pure inertial solution can provide accurate heading information for several hours.

Global Positioning System (GPS)

GPS provides an absolute position measurement using the constellation of GPS satellites. Each satellite monitors its own position and time using transmissions from ground stations located throughout the world. Each satellite then broadcasts its position p_{sat} and the time of transmission t_{sat}. The GPS receiver, aboard the vehicle, forms an equation based on the distance of transmission:

$$c(t - t_{sat}) = \sqrt{(p - p_{sat})^T (p - p_{sat})}, \tag{2-86}$$

where c is the speed of light. Additional satellites provide additional equations, and a full solution for t and p is available with four satellites.

A commercial GPS receiver will typically contain its own Kalman filter that computes position and velocity at periodic intervals. However, some receivers also output the raw measurements.

Additional Measurements

Additional navigation aids are available, but are not used in this application. A star tracker can provide a direct measure of attitude, and is an expensive solution to the attitude drift problem. A magnetic compass can also provide drift-free heading information but is of low quality and requires an Earth magnetic model.

The filter that is developed in this application will also use several "pseudo-measurements". These are measurements that do not originate from actual sensors, but are the result of additional knowledge of the system dynamics. For example, knowledge that the vehicle is fixed to Earth or in exoatmospheric freefall can be interpreted as a measurement, and can effectively be used to calibrate the GPS and IMU sensors.

2.1.6 Initial Alignment

Although an initial condition can be specified, it is desirable and robust to perform initialization based on the first measurements. This technique is very successful when the vehicle is fixed to the Earth at initialization. Under this condition, position can be initialized directly from the GPS and velocity can be computed using $v = \omega_e \times p$, where both position and velocity are expressed in ECI.

Attitude alignment is a more delicate process, but can be performed satisfactorily over large initial angular errors. Two dimensions of attitude alignment can be performed by aligning

the gravity vector, which is measured by the accelerometers. The remaining dimension, which is rotation about the local vertical, can be aligned using a measurement of the Earth's rotation rate. Such alignment usually requires extensive filtering of the gyroscopes, and may not be possible with low quality instrumentation. (A compass can also provide this measurement, but is not pursued here.)

When fixed to Earth, the accelerometer measurement is a combination of gravity and centrifugal acceleration:

$$a_m = B\big(\omega_e \times (\omega_e \times p) - g(p)\big) \tag{2-87}$$

For the purposed of initial alignment, we do not model the measurement error. If the accelerometers are noisy, they should be filtered. An accelerometer bias will affect this initial alignment, but will be treated later in the Kalman filter. The expected measurement is

$$\hat{a}_m = \hat{B}\big(\omega_e \times (\omega_e \times \hat{p}) - g(\hat{p})\big) \approx \hat{B}\big(\omega_e \times (\omega_e \times p) - g(p)\big), \tag{2-88}$$

where we assume that there is little error in the position (from GPS alignment) or in the gravity model. In this regard, any error in the acceleration direction is caused by an incorrect attitude estimate. We seek a corrective rate, to be included in the attitude update equation. For quaternions, this would be:

$$\dot{q} = f_q(q)(\omega + \omega_c). \tag{2-89}$$

A corrective rotation vector that brings $\hat{a}_m$ to a_m can be calculated. First, the angle between vectors can be determined from the vector dot product

$$|\theta_c| = \cos^{-1}\left(\frac{a_m^T \hat{a}_m}{|a_m||\hat{a}_m|}\right). \tag{2-90}$$

Second, the direction of rotation can be represented as a unit vector, using the vector cross product:

$$\frac{\theta_c}{|\theta_c|} = \left(\frac{\hat{a}_m \times a_m}{|\hat{a}_m \times a_m|}\right) \tag{2-91}$$

A gain can be applied to this error and used as the correction rate

$$\omega_c = k_a|\theta_c|\frac{\theta_c}{|\theta_c|} = k_a \cos^{-1}\left(\frac{a_m^T \hat{a}_m}{|a_m||\hat{a}_m|}\right)\left(\frac{\hat{a}_m \times a_m}{|\hat{a}_m \times a_m|}\right). \tag{2-92}$$

Note that the magnitude of the gravity vector is irrelevant for this alignment.

A similar correction rate can be calculated using the rate measurement. An identical process for aligning the Earth rate vector could be formulated, but this would overconstrain the system, providing 4 directions of correction for 3 degrees of freedom. It is preferable to limit the rate alignment to the one remaining dimension. We could instead limit the gravity measurement to one dimension, but this is usually less accurate. While gyroscopes can be very precise, the Earth rate is very small.

The direction of measured acceleration is very close to the local vertical. Therefore, rate measurements will be used to estimate the rotation error about the local vertical. This requires transformation into the NED coordinate system.

The transformation from ECI to NED orientation is done by rotating in longitude and then latitude:

$$T_{ned}(\lambda, L) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos\lambda & 0 & \sin\lambda \\ 0 & 1 & 0 \\ -\sin\lambda & 0 & \cos\lambda \end{bmatrix} \begin{bmatrix} \cos L & -\sin L & 0 \\ \sin L & \cos L & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2-93}$$

The result would actually be in ENU coordinates, so an axis swap is required. Matrix multiplication produces

$$T_{ned}(\lambda, L) = \begin{bmatrix} -\sin\lambda\cos L & \sin\lambda\sin L & \cos\lambda \\ -\sin L & -\cos L & 0 \\ -\cos\lambda\cos L & \cos\lambda\sin L & -\sin\lambda \end{bmatrix}. \tag{2-94}$$

The Earth rate vector in ECI [14],

$$\omega_e = \begin{pmatrix} 0 & 0 & 7.2921150\times10^{-5} \end{pmatrix}^T \text{ rad/sec}, \tag{2-95}$$

transforms to

$$\omega_{ned} = T_{ned}\omega_e = \begin{pmatrix} \cos\lambda \\ 0 \\ -\sin\lambda \end{pmatrix} |\omega_e|. \tag{2-96}$$

If we assume that the body frame is misaligned by an angle ψ_{ned} of pure NED yaw, the value measured by the gyroscopes will be:

$$\omega_m = \begin{bmatrix} \cos\psi_{ned} & -\sin\psi_{ned} & 0 \\ \sin\psi_{ned} & \cos\psi_{ned} & 0 \\ 0 & 0 & 1 \end{bmatrix} T_{ned}\omega_e = \begin{pmatrix} \cos\psi_{ned}\cos\lambda \\ \sin\psi_{ned}\cos\lambda \\ -\sin\lambda \end{pmatrix} |\omega_e|. \tag{2-97}$$

Given this measurement, the error can be calculated from the first two terms:

$$\psi_{ned} = \tan^{-1}\left(\frac{\omega_m(2)}{\omega_m(1)}\right). \tag{2-98}$$

The four quadrant arctangent should be used to accommodate large errors. Alternatively, the arctangent can be linearized if small errors are expected. The measured rate should be heavily filtered before the arctangent.

We now add a scalar gain and convert from NED to body coordinates,

$$\omega_c = -k_\psi B_{ned}\psi_{ned}\,, \tag{2-99}$$

and we have an additional rate correction for the quaternion update. The DCM B_{ned} is the rotation from NED to body coordinates. It can be expressed in terms of the B, the ECI DCM, using

$$B = B_{ned}T_{ned}(\lambda, L)\,. \tag{2-100}$$

Inverting produces

$$B_{ned} = BT_{ned}^T(\lambda, L)\,. \tag{2-101}$$

Note that this technique will fail for initialization near the poles, where the Earth rate and gravity vectors are coincident.

The equations of navigation will be central to the design of this navigation system. However, we also need to develop formal treatment for estimation under uncertainty. For this purpose, we will now shift discussion to the Kalman Filter.

2.2 The Kalman Filter

The Kalman filter is widely used in navigation and has proven to be successful in these applications. It presents a systematic approach for estimation in the presence of uncertainty. It is directly applicable to Linear Time-Varying (LTV) systems and can be extended to lightly nonlinear systems. The filter itself is linear, allowing for reasonable computation and implementation requirements. Under certain assumptions, the Kalman filter provides the optimal estimate for a given noise environment, which makes it superior to many types of estimators.

It is important to discuss these assumptions because they characterize the filter's strengths and limitations. Though in practical implementation the Kalman filter is rarely actually optimal, the filter performs very well in many situations. In system modeling and design, we will strive to best meet these assumptions, and will take caution if they are violated.

The first assumption is that the system in linear. This includes LTV systems, which makes the Kalman filter more powerful than most linear observers. As we will explore later, this also allows for approximate treatment of weakly nonlinear systems, in that such systems can usually be approximated by LTV systems.

The second assumption is that the system uncertainty is white noise, meaning that the noise statistics are independent of previous noise values. This assumption would be restrictive, but techniques exist for removing noise correlation by augmenting the deterministic system model.

Under the previous two assumptions, the Kalman filter provides the estimate with the minimum unconditioned error variance of all *linear* estimators. This is acceptable for most applications, because *nonlinear* estimators maybe difficult to formulate or computationally intensive. Note that we are discussing the linearity of the estimator and are still assuming an LTV system.

With an additional assumption that the noise is Gaussian, the Kalman filter becomes the optimal estimator by most reasonable criteria. In this case, the Kalman estimate is equivalent to the conditional expectation of the state x given all previous measurements z, $\hat{x}(t) = E\{x(t) \mid z(t)\}$, and the conditional variance is minimized. Mathematically, the Gaussian assumption allows for direct expression of the system probability distribution function, which facilitates the proof and leads to a linear form [3, pp. 228-232].

The Gaussian assumption is not met in general, but many types of noise are approximately Gaussian. The Central Limit Theorem states that the probability distribution for the sum of many independent random variables approaches a Gaussian distribution. Also, the actual higher order noise statistics are rarely known, so the Gaussian assumption may be the best choice. For LTV systems, the success of this assumption seriously limits the benefit of pursuing nonlinear estimators. For our derivation, we will not depend on this assumption, but the performance of the Kalman filter under Gaussian noise is an important strength.

An additional tacit assumption is that the LTV model is an accurate representation of the system, both in terms of parameter values and in the underlying model structure. This will be of particular concern when we linearize a nonlinear system for use with the Kalman filter. It is inevitable that modeling errors exist at some scale, and the effect of such errors will be analyzed.

Alternatives to the Kalman Filter

The most notable shortcoming of the Kalman filter is that it is derived for LTV systems. Although the filter can be extended to accommodate nonlinear systems, there is no guarantee of optimality or even stability. While performance has been successful in many nonlinear applications, convergence analysis is exceedingly difficult.

However, any weaknesses in the Kalman filter should be viewed in comparison with the alternatives. Linearization is a crucial technique for many forms of engineering analysis. Full treatment of nonlinear systems is difficult and there are only a few successful approaches.

This first alternative is a nonlinear observer. One example can be found in [22]. Of course the Kalman filter is an observer, but it does not guarantee convergence when applied to nonlinear systems. Instead, we could pursue a nonlinear observer of a simpler form with the goal of proving global exponential stability. However, the simplification of the gain law inevitably discards the linear optimal techniques of the Kalman filter. This exchange sacrifices many of the benefits in noise discrimination and observation of LTV systems.

The second alternative is a nonlinear optimal filter, but this proves to be impossible from a practical standpoint. The optimal nonlinear estimate can be formulated in terms of the probability density functions of the state and measurement [7, 10]. However, the entire state probability density function must be propagated with time. The covariance estimate of the Kalman filter is replaced by a continuum of probability estimates. The resulting differential equation, with nonlinear state dynamics, is an intractable mathematical problem. Even a numerical solution requires a prohibitively large number of variables to approximate the probability density function.

A third alternative is an adaptive filter. This refers to a broad class of systems that use adaptive modeling techniques to intelligently estimate the state and noise dynamics. Many can be described as extensions to the Kalman filter. The Sage-Husa filter [16], allows for unknown noise, and estimates the noise statistics along with the state. Others methods use online tuning of the Kalman Filter, such as [17]. Some of the noise models described in Sections 2.2.5.2 and 2.3.1.3 can be adapted in real time. While these techniques are arguably more sophisticated than the Kalman filter and may provide increased performance, they do not directly address the nonlinear stability problem.

A second variety of adaptive techniques exist that attempt to estimate the entire process model. Here the entire system is parameterized, using poles and zeros, wavelets, neural networks, or other model sets. The physics of the process is not necessarily considered. This presents a fatal weakness, because the physics of navigation is highly known and provides

significantly more information than the sensors alone. However, these techniques can be very useful for characterizing the unknown parts of the dynamics, such as the disturbance model.

2.2.1 Continuous-Discrete Formulation

The Kalman filter derived here is a hybrid between the continuous-time and discrete-time formulations. With this technique, the system state and covariance are propagated continuously between measurements, but are updated extemporaneously when each measurement arrives.

Most physical phenomena are inherently continuous-time, such as rigid body dynamics and the force of gravity. Even the IMU bias dynamics, which have an effect only at the discrete times of measurement, are samples of an underlying continuous physical process, such as the thermal behavior of the sensor. Discrete equivalents are approximate and can be cumbersome to calculate in real time for an LTV system. The continuous Kalman filter uses the continuous equations directly, and can be implemented using a numerical differential equation solver. This solver must operate at discrete time steps, but can provide an accurate solution to the continuous differential equations.

Measurements, however, occur at discrete times, either because a continuous-time sensor is sampled or because the sensor itself is discrete-time. While a sample-and-hold architecture can be used to provide a continuous-time measurement for use in the continuous-time Kalman filter, this is no longer optimal because it gives weight to the measurement for an extended time period. Technically, the measurement is valid only at the instant of measurement. It is therefore preferable to process measurements directly in discrete time.

This hybrid filter conveniently allows for multirate and sporadic measurements. Measurement updates can be performed with whatever measurements are available, or omitted entirely. This provides natural handling of measurement dropouts and produces a state estimate that is available between measurements.

2.2.1.1 Problem Statement and Assumptions

The system is modeled as a continuous LTV system

$$\dot{x}(t) = A(t)x(t) + u(t) + F(t)w(t) \tag{2-102}$$

where the continuous state x is driven by a known control input u and unknown process noise w, and $A(t)$ and $F(t)$ are matrices that may be time-varying. Measurements z are taken at discrete times t_k and contain unknown discrete-time measurement noise n:

$$z(t_k) = H(t_k)x(t_k) + D(t_k)u(t_k) + L(t_k)n(t_k)\,. \tag{2-103}$$

The arguments t and t_k will usually be omitted for clarity, but the explicit dependence on time should be noted. We will also sometimes write $z[k]$ or $n[k]$ for discrete-time quantities, where $t_k = kT$ and T is the sample time.

For this derivation, we assume zero mean noise:

$$E\{w\} = 0 \tag{2-104}$$
$$E\{n\} = 0\,. \tag{2-105}$$

We also assume that the process and measurement noise are white (time-uncorrelated):

$$E\{w(t)w(t-\tau)\} = Q\delta_c(\tau) \tag{2-106}$$
$$E\{n[k]n[k-m]\} = R\delta[m]\,, \tag{2-107}$$

where $\delta_c(\tau)$ is the continuous Dirac delta function (unit impulse) and $\delta[m]$ is the discrete unit impulse. The variables τ and m are arbitrary offsets. A technique for removing time correlation by augmenting the system model is presented in Section 2.2.5.2. With this method, it is usually possible to satisfy this assumption.

Finally, for ease in derivation, we assume that the process and measurement are uncorrelated:

$$E\{w(t)n[k]\} = 0 \quad \forall t,k\,. \tag{2-108}$$

Correlation between the process and measurement noise can be included in the derivation, but is not needed for this application.

We seek the "best" state estimate $\hat{x}$. We will discuss optimality with various cost functions, and ultimately restrict the result to linear estimators.

2.2.1.2 Minimum Conditional Error Variance

The strongest condition for optimality is to minimize the variance of the estimation error conditional on the available measurements. This derivation, taken from [7, p. 104], shows that the resulting estimate is the conditional expectation.

The conditional probability of the state x given z, can be represented by the probability density function

$$p(x|z) = \frac{p(x \cap z)}{p(z)} \tag{2-109}$$

where p here is not to be confused with the position vector p used in other sections. This is the ratio of the joint probability of x and z to the probability of z.

The conditional expectation is then defined as the probability weighted average of x over the entire state space region R^n:

$$E\{x|z\} = \int_{R^n} xp(x|z)dx\,. \tag{2-110}$$

The estimation error

$$e = \hat{x} - x\,, \tag{2-111}$$

can be used to define a weighted square error loss function

$$v(e) = e^T We\,, \tag{2-112}$$

where W is a constant symmetric positive definite matrix. Note that this loss function has a global minimum at the origin and derivative

$$\frac{dv}{de} = 2We = 2W(\hat{x} - x)\,. \tag{2-113}$$

The conditional total cost is then given by weighting this loss function by the conditional probability density function and integrating over R^n:

$$V = \int_{R^n} v(e)\,p(x|z)dx\,. \tag{2-114}$$

The minimum occurs when

$$\frac{dV}{d\hat{x}} = \int_{R^n} \frac{dv}{de}\frac{de}{d\hat{x}}\,p(x|z)dx = 2W\int_{R^n} ep(x|z)dx = 0\,, \tag{2-115}$$

which requires that

$$\int_{R^n} (\hat{x} - x)p(x|z)dx = 0\,. \tag{2-116}$$

The estimate $\hat{x}$ can come out of the integral and the equation can be rewritten as

$$\hat{x}\int_{R^n} p(x|z)dx = \int_{R^n} xp(x|z)dx\,. \tag{2-117}$$

Since for any probability density function

$$\int_{R^n} p(x|z)dx = 1\,, \tag{2-118}$$

this expression simplifies to

$$\hat{x} = \int_{R^n} xp(x|z)dx = E\{x|z\}, \tag{2-119}$$

which gives the promised result that the optimal estimate is the conditional expectation. Surprisingly, this result is independent of the weighting matrix *W*. This result can be expanded to include a wide variety of loss functions, under certain assumptions regarding the probability density function [11]. The optimality of filters based on this result is therefore very general and broad.

Direct evaluation of (2-119) may be possible using Bayes Theorem

$$p(x|z) = \frac{p(z|x)p(x)}{p(z)}. \tag{2-120}$$

The difficulty is that $p(x)$ and $p(x|z)$ must be propagated through the state dynamics. This is unfeasible for most probability distributions and for nonlinear systems. However, the case of Gaussian noise can be solved because an LTV system preserves Gaussian statistics. The result is linear in form and is identical to the Kalman filter. This derivation of the Kalman filter is explored in [3] and, of course, by Kalman himself in [11].

2.2.1.3 Minimum Unconditional Error Variance

An alternative choice of the cost function is the *unconditional* expectation of the weighted square error:

$$V = \int_{R^n} e^T We p(x)dx = E\{e^T We\}. \tag{2-121}$$

Note that the unconditional probability $p(x)$ now replaces $p(x|z)$. In this fashion, the actual measurements have been removed from the cost function, but they are still used by the estimator to best minimize this new cost. With no weighting, $W=I$, this cost is the *unconditional* variance of the error. The weighting matrix *W* is included for completeness, but is once again found to have no effect.

In general, cost minimization occurs when

$$\frac{dV}{d\hat{x}} = E\{2We\} = 2WE\{\hat{x} - x\} = 0, \tag{2-122}$$

which occurs when

$$E\{\hat{x}\} = E\{x\}, \tag{2-123}$$

showing that the optimal estimate is unbiased. We will proceed to develop a recursive formula for the optimal estimator. This derivation was greatly influenced by [5] and [13] for continuous systems and [3] for discrete systems. The result here is specific for continuous-discrete systems and adds the effect of the weighting matrix *W*.

In order to estimate the state, the filter will also estimate the error covariance

$$P = E\left\{ee^T\right\}. \tag{2-124}$$

Note that this is a covariance matrix only when the error is zero mean, which occurs because of (2-123).

Initialization

Since optimality requires an unbiased estimate, recursion must begin at

$$\hat{x}_o = E\left\{x(t_o)\right\} = \bar{x}_o. \tag{2-125}$$

The corresponding initial covariance is

$$P_o = E\left\{(\hat{x}_o - x_o)(\hat{x}_o - x_o)^T\right\} = E\left\{(\bar{x}_o - x_o)(\bar{x}_o - x_o)^T\right\} = P_{x_o}, \tag{2-126}$$

which is the covariance of the initial state x_o.

Propagation between Measurements

In the absence of measurements, the unbiased estimate is simply the expected propagation:

$$\dot{\hat{x}} = E\left\{\dot{x}\right\} = E\left\{A(t)x + u + F(t)w(t)\right\} = A(t)\hat{x} + u. \tag{2-127}$$

The error covariance propagation can be calculated:

$$\dot{P} = \frac{d}{dt}E\left\{ee^T\right\} = E\left\{\frac{d}{dt}ee^T\right\} = E\left\{\dot{e}e^T + e\dot{e}^T\right\}. \tag{2-128}$$

Using the estimated and actual state dynamics, the error dynamics are given by

$$\dot{e} = \dot{\hat{x}} - \dot{x} = A(t)\hat{x} + u - A(t)x - u - F(t)w(t) = A(t)e - F(t)w(t), \tag{2-129}$$

which leads to

$$E\left\{\dot{e}e^T + e\dot{e}^T\right\} = E\left\{Aee^T - Fwe^T + ee^T A^T - ew^T F^T\right\}, \tag{2-130}$$

where $E\{ee^T\} = P$, but $E\{Fwe^T\}$ requires further attention. From (2-129), the error solution can be expressed as a convolution with the error state transition matrix:

$$e = -\int_{t_o}^{t} \Phi_e(t,\tau)F(\tau)w(\tau)d\tau\,. \tag{2-131}$$

In this discussion, the variable τ is a dummy integration variable, and does not refer to a torque input. Using this expression yields

$$Fwe^T = -F(t)w(t)\int_{t_o}^{t} w^T(\tau)F^T(\tau)\Phi_e^T(t,\tau)d\tau = -\int_{t_o}^{t} F(t)w(t)w^T(\tau)F^T(\tau)\Phi_e^T(t,\tau)d\tau\,, \tag{2-132}$$

where the functions of the variable *t* can be brought inside the integral of variable τ. Taking the expectation and using the noise characteristics of *w* produces

$$E\{Fwe^T\} = -\int_{t_o}^{t} F(t)Q\delta_c(t-\tau)F^T(\tau)\Phi_e^T(t,\tau)d\tau\,, \tag{2-133}$$

causing the integrand to be nonzero only at $t = \tau$, where $\Phi_e^T(t,t) = I$. Careful consideration of the limit shows that exactly half of the Dirac pulse is within the integral bounds [13, p. 164], yielding

$$E\{Fwe^T\} = -\frac{1}{2}F(t)QF^T(t). \tag{2-134}$$

This is symmetric, so $E\{ew^TF^T\} = E\{Fwe^T\}$. Substituting these results back into (2-130) yields

$$\dot{P} = AP + PA^T + FQF^T\,, \tag{2-135}$$

which gives the continuous error covariance propagation in the absence of measurements.

Measurement Update

We now consider the update that occurs when a measurement arrives. Since this update takes place instantaneously, the subscripts '-' and '+' will be used to distinguish values before (a priori) and after (a posteriori) the measurement update, respectively. This gives the definitions of $\hat{x}_-$, e_-, and P_-, and $\hat{x}_+$, e_+, and P_+. We will also sometimes explicitly include time by writing expression such as $\hat{x}(t_-)$ and $\hat{x}(t_+)$.

In order to proceed, we now limit ourselves to *linear* estimators. In doing so, we are finding the optimal linear minimum variance estimate.

The most general form of a linear estimator would be a linear combination of the all information:

$$\hat{x}_+ = K_1\hat{x}_- + K_2 z + K_3 u\,. \tag{2-136}$$

The optimal estimate must be unbiased. Substituting for *z* and taking the expectation yields

$$E\{\hat{x}_+\} = E\{K_1\hat{x}_- + K_2(Hx + Du + Ln) + K_3 u\}\,. \tag{2-137}$$

Since the a priori estimate was unbiased, $E\{\hat{x}_-\} = E\{x\}$. Using also $E\{n\} = 0$ and $E\{u\} = u$, this reduces to

$$E\{\hat{x}_+\} = (K_1 + K_2 H)E\{x\} + (K_2 D + K_3)u\,. \tag{2-138}$$

For independence to the control input, we must have

$$K_2 D + K_3 = 0\,. \tag{2-139}$$

For an unbiased a posteriori estimate, we require

$$K_1 + K_2 H = I\,. \tag{2-140}$$

If we define $K = K_2$, then (2-139) and (2-140) become $K_1 = I - KH$ and $K_3 = -KD$. The resulting expression is that of a linear observer:

$$\hat{x}_+ = (I - KH)\hat{x}_- + Kz - KDu = \hat{x}_- + K(z - H\hat{x}_- - Du) = \hat{x}_- + K(z - \hat{z}), \tag{2-141}$$

which forms a correction based on the measurement error.

The optimal gain *K* that minimizes *V* can now be found by evaluating the following derivative expression:

$$\frac{dV}{dK} = \frac{d}{dK}E\{e^T W e\} = E\left\{\frac{d}{dK}e^T W e\right\} = 0\,. \tag{2-142}$$

This equation describes the derivative of a scalar with respect to every element in a matrix. This derivative can be organized in a matrix of the same size as *K* with components

$$\left(\frac{dV}{dK}\right)_{ij} = \frac{dV}{dK_{ij}}\,. \tag{2-143}$$

Several matrix derivative formulae follow from this definition. First, the derivative of a matrix norm can be expressed by the components

$$\left(\frac{d}{dK}a^T Kb\right)_{ij} = \frac{d}{dK_{ij}}a^T Kb = \frac{d}{dK_{ij}}a_i K_{ij} b_j = a_i b_j, \tag{2-144}$$

because each derivative has only one nonzero term. This can be written compactly as the outer product

$$\frac{d}{dK}a^T Kb = ab^T. \tag{2-145}$$

Transposing K will transpose the derivative:

$$\frac{d}{dK}a^T K^T b = \left(\frac{d}{dK^T}a^T K^T b\right)^T = \left(ab^T\right)^T = ba^T. \tag{2-146}$$

Finally, the derivative of the following weighted quadratic in K can be differentiated one K at a time using the previous results:

$$\frac{d}{dK}a^T K^T WKb = \left(WKb\right)a^T + \left(WKa\right)b^T = WK\left(ba^T + ab^T\right), \tag{2-147}$$

for symmetric W. If $a = b$, this reduces to

$$\frac{d}{dK}a^T K^T WKa = 2WKaa^T. \tag{2-148}$$

We can now proceed to build an expression for V and evaluate the derivative. Using the observer form given by (2-141), the a posteriori error can be related to the a priori error

$$e(t_+) = \hat{x}(t_+) - x(t) = \hat{x}(t_-) + K(t)\left(H(t)x(t) + L(t)n(t) - H(t)\hat{x}(t_-)\right) - x(t) \tag{2-149}$$

or

$$e_+ = \left(I - K(t)H(t)\right)e_- + KLn(t). \tag{2-150}$$

Taking the weighted inner product, and combining scalars with their transposes, yields

$$\begin{aligned} e_+^T We_+ &= \left(\left(I - KH\right)e_- + KLn\right)^T W\left(\left(I - KH\right)e_- + KLn\right) \\ &= e_-^T\left(I - KH\right)^T W\left(I - KH\right)e_- + e_-^T\left(I - KH\right)^T WKLn \\ &\quad + n^T L^T K^T W\left(I - KH\right)e_- + n^T L^T K^T WKLn \\ &= e_-^T e_- - 2e_-^T WKHe_- + e_-^T H^T K^T WKHe_- + 2e_-^T WKLn \\ &\quad - 2n^T L^T K^T WKHe_- + n^T L^T K^T WKLn. \end{aligned} \tag{2-151}$$

The derivative can now be taken using the previous formulae, with terms such as e_-, We_-, He_-, and Ln forming the a and b vectors:

$$\frac{d}{dK}e_+^T We_+ = 2W\left(-e_-e_-^T H^T + KHe_-e_-^T H^T + e_-n^T L^T - K\left(He_-n^T L^T + Lne_-^T H^T\right) + KLnn^T L^T\right). \quad (2\text{-}152)$$

We will prepare to take the expectation by noting that

$$E\left\{e_-e_-^T\right\} = P_- \quad (2\text{-}153)$$

$$E\left\{nn^T\right\} = R. \quad (2\text{-}154)$$

Also, the a priori error e_- is not influenced by the the current noise $n = n(t_k)$, so we have

$$E\left\{e_-n^T\right\} = E\left\{ne_-^T\right\} = 0. \quad (2\text{-}155)$$

Proceeding with the expectation yields

$$\frac{dV}{dK} = E\left\{\frac{d}{dK}e^T e\right\} = 2W\left(-P_-H^T + KHP_-H^T + KLRL^T\right) = 0 \quad (2\text{-}156)$$

The weighting matrix drops out, and the optimal gain K is governed by

$$K\left(HP_-H^T + LRL^T\right) = P_-H^T. \quad (2\text{-}157)$$

By the definition of covariance, the matrices P and R are both symmetric positive definite and inversion is therefore possible:

$$K = P_-H^T\left(HP_-H^T + LRL^T\right)^{-1}. \quad (2\text{-}158)$$

This matrix inversion is computationally intensive, and an alternative technique is presented in Section 2.2.5.3.

With this optimal gain, an update law for P is required. This can be found using (2-150):

$$P_+ = E\left\{e_+e_+^T\right\} = E\left\{\left((I-KH)e_- + KLn\right)\left((I-KH)e_- + KLn\right)^T\right\}. \quad (2\text{-}159)$$

Multiplying and taking the expectation produces

$$P_+ = (I-KH)P_-(I-KH)^T + KLRL^T K^T. \quad (2\text{-}160)$$

This is known as Joseph's Stabilized Form [8, p. 233] and has desirable numerical properties because of its symmetry.

Equation (2-160) can be simplified by multiplying out first term:

$$P_+ = P_- - KHP_- - P_-H^TK^T + KHH^TK^T + KLRL^TK^T, \quad (2\text{-}161)$$

and substituting the optimal gain relation of (2-148):

$$P_+ = P_- - KHP_- - K\left(HP_-H^T + LRL^T\right)K^T + KHH^TK^T + KLRL^TK^T = \left(I - KH\right)P_-. \quad (2\text{-}162)$$

Summary

The filter requires the initialization $\hat{x}_o$, P_o and specification of the noise covariances Q and R, which may be time-varying. Then the state estimate and covariance are propagated continuously between measurements using

$$\dot{\hat{x}} = A\hat{x} + u \quad (2\text{-}163)$$
$$\dot{P} = AP + PA^T + FQF^T \quad (2\text{-}164)$$

and updated discontinuously at measurements using

$$K = P_-H^T\left(HP_-H^T + LRL^T\right)^{-1} \quad (2\text{-}165)$$
$$\hat{x}_+ = \hat{x}_- + K\left(z - H\hat{x}_-\right) \quad (2\text{-}166)$$
$$P_+ = \left(I - KH\right)P_-. \quad (2\text{-}167)$$

2.2.2 Nonlinear Extension

Consider the continuous nonlinear system with discrete measurements given by

$$\dot{x}(t) = f\left(x(t), u(t), w(t), t\right) \quad (2\text{-}168)$$
$$z(t_k) = h\left(x(t_k), u(t_k), n(t_k), t_k\right) \quad (2\text{-}169)$$

A general optimal estimator for this system is a difficult mathematical problem. The primary obstacle is the propagation of the error statistics which now involve the full probability density functions. The estimation error is no longer defined by its covariance, because even simple noise inputs are complicated by the nonlinear dynamics.

The alternative approach is to linearize the dynamics, and then apply the Kalman filter. This linearization can be done with respect to a predefined trajectory, or with respect to the estimated trajectory. This first method is suitable for repeated motion, such as manufacturing tools, where the trajectory is well known in advance. For navigation, the flight path is usually not well known, so we use the latter approach, which is commonly known in the literature as the Extended Kalman Filter (EKF).

The EKF raises issues of stability. The feedback of the estimate into the linearization creates a new nonlinearity which can have adverse effects. The EKF will be successful if the model

is sufficiently smooth in the region of operation to be well approximated by linear functions. The topics of convergence and stability will be addressed in Section 2.2.4.4.

2.2.2.1 Linearization

The standard approach to the Extended Kalman Filter is to replace all matrices in the Kalman Filter with their corresponding Jacobians. For example:

$$A(t) \approx \left.\frac{\partial f}{\partial x}\right|_{\hat{x},t} \tag{2-170}$$

and similarly for the matrices *F*, *H*, *D*, and *L*. Each partial derivative is evaluated at the estimated state. This approximation is justified from a Taylor series expansion. However, we will take a more deliberate approach, and in the process uncover some techniques that will simplify computation and make the result more linear.

We begin by dividing the state into a nonlinear estimate and a small linear estimation error:

$$x(t) = \hat{x}(t) + \delta x(t)\,. \tag{2-171}$$

The linear state will be estimated by the Kalman filter, while the nonlinear state will be propagated independently. A logical choice for the nonlinear propagation is the differential equation

$$\dot{\hat{x}} = f(\hat{x}, u, w = 0, t)\,, \tag{2-172}$$

with discontinuous updates when measurements arrive

$$\hat{x}_+ = \hat{x}_- + i\,. \tag{2-173}$$

The term *i* in this Extended Kalman Filter is analogous to the innovations process of the Kalman filter. The innovations process represents the updates to the state estimate, and should be zero mean white noise under optimal filter operation [5, pp. 425-427]. In the EKF, *i* is a variable of our choice, and we will use it to offload the linear error state onto the full nonlinear state estimate.

The previous choices for nonlinear propagation leave the following dynamics of the estimation error:

$$\dot{\delta x} = \dot{x} - \dot{\hat{x}} = f(x, u, w, t) - f(\hat{x}, u, w = 0, t) \tag{2-174}$$

$$\delta\hat{x}_+ = \delta\hat{x}_- + \left(x - \hat{x}_+\right) - \left(x - \hat{x}_-\right) = \delta\hat{x}_- - i \tag{2-175}$$

If we now assume that δx and w are small, a linear Taylor series expansion gives

$$\delta\dot{x} \approx \left(\left. \frac{\partial f}{\partial x} \right|_{\hat{x},u,w=0,t} \right) \delta x + \left(\left. \frac{\partial f}{\partial w} \right|_{\hat{x},u,w=0,t} \right) w \,. \tag{2-176}$$

We can now begin to build a Kalman filter to estimate the estimation error $\delta\hat{x}$ using

$$A(t) = \left. \frac{\partial f}{\partial x} \right|_{\hat{x},u,w=0,t} \tag{2-177}$$

$$F(t) = \left. \frac{\partial f}{\partial w} \right|_{\hat{x},u,w=0,t} \tag{2-178}$$

$$u = 0 \,. \tag{2-179}$$

Here u is the control input for the Kalman filter, while the control input for the nonlinear propagation can still be nonzero. Additional linearization is required for the measurement, which can be expressed using the estimation error

$$z(t_k) = h\big(x(t_k), u(t_k), n(t_k), t_k\big) = h\big(\hat{x} + \delta x, u, n, t_k\big). \tag{2-180}$$

We now divide our measurement into nonlinear and small error components

$$z(t_k) = \hat{z}(t_k) + \delta z(t_k) \tag{2-181}$$

where the predicted measurement

$$\hat{z}(t_k) = h\big(\hat{x}, u, n = 0, t_k\big) \tag{2-182}$$

leaves a small error

$$\delta z(t_k) = z(t_k) - \hat{z}(t_k) = h\big(\hat{x} + \delta x, u, n, t_k\big) - h\big(\hat{x}, u, n = 0, t_k\big). \tag{2-183}$$

This result can be linearized, assuming small δx and n:

$$\delta z \approx \left(\left. \frac{\partial h}{\partial x} \right|_{\hat{x},u,n=0,t} \right) \delta x + \left(\left. \frac{\partial h}{\partial n} \right|_{\hat{x},u,n=0,t} \right) n \tag{2-184}$$

This now gives the linear measurement and measurement noise matrices for our Kalman filter:

$$H(t) = \left. \frac{\partial h}{\partial x} \right|_{\hat{x},u,n=0,t} \tag{2-185}$$

$$L(t) = \left. \frac{\partial h}{\partial n} \right|_{\hat{x},u,n=0,t} \,. \tag{2-186}$$

2.2.2.2 Measurement Pre-Compensation

The technique presented here provides a treatment for nonlinear measurements that is superior in many situations. The strategy is to remove the nonlinearity before the measurement is given to the Kalman filter. In effect, the nonlinearity of a measurement or group of measurements is inverted to produce a linear measurement. In many cases, the result is a more accurate linearization for the Kalman filter and a simpler expression for the measurement Jacobian.

For example, consider the following nonlinear measurement:

$$z = (x + n)^3 . \tag{2-187}$$

The nonlinear function can be inverted, and a new measure defined:

$$z' = \sqrt[3]{z} = x + n \tag{2-188}$$

Subtracting the predicted measurement produces a small measurement error

$$\delta z = z' - \hat{x} = \delta x + n \tag{2-189}$$

which is linear in the error state and noise. In this example, a direct linearization would have produced

$$\delta z = 3\hat{x}^2 (\delta x + n). \tag{2-190}$$

This example was particularly simple because the noise was inside the nonlinear function and there was only one state involved. In general, measurement pre-compensation will still require linearization and the inverse nonlinearity must be computed at the estimated state.

This technique can be formalized and the measurement Jacobian can be modified to include the effect of pre-compensation. Although the notion of inverting a nonlinearity is fairly common, this derivation is original. For the specific equations of this application, this technique leads to simplified equations and more accurate linearization.

Consider the case where a block of measurements z_b depends primarily on a corresponding block of states x_b:

$$z_b = h_b(x_b; x_{\sim b}, u, n_b, t) = h_b(\hat{x}_b + \delta x_b; \hat{x}_{\sim b} + \delta x_{\sim b}, u, n_b, t), \tag{2-191}$$

where $x_{\sim b}$ are the additional states of secondary dependence. For example, a gyroscope measures body rates primarily, but is affected by biases and scale factors which can be included as other state variables. It is frequently possible that this portion of the measurement function can be inverted and the block states x_b recovered using

$$x_b = h_b^{-1}(z_b; x_{\sim b}, u, n_b, t) \tag{2-192}$$

Now define a new measurement that is compensated by the estimated state

$$z_b' = h_b^{-1}\left(h_b\left(x_b; x_{\sim b}, u, n_b, t\right); \hat{x}_{\sim b}, u, n_b = 0, t\right). \quad (2\text{-}193)$$

This can be rewritten using the estimation errors

$$z_b' = h_b^{-1}\left(h_b\left(\hat{x}_b + \delta x_b; \hat{x}_{\sim b} + \delta x_{\sim b}, u, n_b, t\right); \hat{x}_{\sim b}, u, n_b = 0, t\right). \quad (2\text{-}194)$$

Assuming once again that δx and n_b are small, the linear Taylor series approximation can be used, first on the inner function:

$$z_b' \approx h^{-1}\left(h\left(\hat{x}_b; \hat{x}_{\sim b}, u, n_b = 0, t\right) + \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_b, \hat{x}_{\sim b}, u, n_b = 0, t} \delta x + \left.\frac{\partial h}{\partial n_b}\right|_{\hat{x}_b, \hat{x}_{\sim b}, u, n_b = 0, t} n_b; \hat{x}_{\sim b}, u, n = 0, t \right), \quad (2\text{-}195)$$

and then again on the inverse function:

$$z_b' \approx \hat{x}_b + \left.\frac{\partial h^{-1}}{\partial z_b}\right|_{\hat{z}_b, \hat{x}_{\sim b}, u, n_b = 0, t} \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_b, \hat{x}_{\sim b}, u, n_b = 0, t} \delta x + \left.\frac{\partial h^{-1}}{\partial z_b}\right|_{\hat{z}_b, \hat{x}_{\sim b}, u, n_b = 0, t} \left.\frac{\partial h}{\partial n_b}\right|_{\hat{x}_b, \hat{x}_{\sim b}, u, n_b = 0, t} n_b, \quad (2\text{-}196)$$

where the inverse function is evaluated at the expected measurement

$$\hat{z}_b = h_b\left(\hat{x}_b; \hat{x}_{\sim b}, u, n_b = 0, t\right). \quad (2\text{-}197)$$

The small measurement for this block can then be defined as

$$\delta z_b \approx z'_b - \hat{x}_b = \left.\frac{\partial h^{-1}}{\partial z_b}\right|_{\hat{z}_b, \hat{x}_{\sim b}, u, n_b = 0, t} \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_b, \hat{x}_{\sim b}, u, n = 0, t} \delta x + \left.\frac{\partial h^{-1}}{\partial z_b}\right|_{\hat{z}_b, \hat{x}_{\sim b}, u, n_b = 0, t} \left.\frac{\partial h}{\partial n}\right|_{\hat{x}_b, \hat{x}_{\sim b}, u, n, t} n_b. \quad (2\text{-}198)$$

These components can then be placed into the Kalman filter. The corresponding rows of the *H* matrix are

$$H_b(t) = \left.\frac{\partial h^{-1}}{\partial z_b}\right|_{\hat{z}_b, \hat{x}_{\sim b}, u, n_b = 0, t} \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_b, \hat{x}_{\sim b}, u, n = 0, t} \quad (2\text{-}199)$$

and the noise portion of *L* for these block measurements is

$$L_b(t) = \left.\frac{\partial h^{-1}}{\partial z_b}\right|_{\hat{z}_b, \hat{x}_{\sim b}, u, n_b = 0, t} \left.\frac{\partial h}{\partial n}\right|_{\hat{x}_b, \hat{x}_{\sim b}, u, n = 0, t}. \quad (2\text{-}200)$$

The advantage of measurement pre-compensation is that the nonlinearity is removed as much as possible. Note that the partial derivatives of both *h* and h^{-1} appear in the Jacobians. This is because the inverse is evaluated at the estimated state.

2.2.2.3 Offloading the Estimation Error

For the previous assumptions to hold, a small estimation error must be maintained. Various techniques are available for offloading the estimation error. The most convenient is to initialize and maintain zero error:

$$\delta\hat{x}(t) = 0 \quad \forall t \,. \tag{2-201}$$

As the Kalman filter minimizes the variance

$$E\left\{e^T e\right\} = E\left\{\left(\delta\hat{x} - \delta x\right)^T \left(\delta\hat{x} - \delta x\right)\right\}, \tag{2-202}$$

and is unbiased, this will maintain the smallest δx possible. The continuous-time error propagation is now trivial:

$$\dot{\delta\hat{x}} = A\delta\hat{x} + u = 0 \,. \tag{2-203}$$

The measurement update is

$$\delta\hat{x}_+ = \delta\hat{x}_- - i + K\left(\delta z - H\delta\hat{x}_-\right) = \delta\hat{x}_- - i + K\delta z \,, \tag{2-204}$$

and, to maintain $\delta\hat{x}_+ = 0$, we must choose

$$i = K\delta z \,. \tag{2-205}$$

In other words, we pass the innovation directly from the Kalman filter to the nonlinear estimate. Note that the error state matrices are still required to propagate the covariance and calculate the Kalman gain:

$$\dot{P} = AP + PA^T + FQF^T \tag{2-206}$$

$$K = P_- H^T \left(HP_- H^T + R\right)^{-1} \tag{2-207}$$

$$P_+ = \left(I - KH\right)P_- \,. \tag{2-208}$$

This choice of maintaining zero error simplifies the Jacobians. Any approximation for small quantities, such as small angles or dropping higher order terms, becomes exact when the error state is zero. As a result, these approximations can be used freely without degrading the estimation accuracy.

2.2.3 Data Forgetting

As the optimal solution, the Kalman filter necessarily uses all past data that is available. This apparent strength for measurement fusion becomes a weakness in the presence of modeling inaccuracies. As time progresses, the estimated covariance may diminish, reducing the Kalman gain, while the actual error may increase. The result is a sluggish filter that stubbornly clings to its incorrect estimate.

There are several solutions to this problem, presented in [7, pp.279-288] and [13, pp. 218-222]. Most techniques sacrifice optimality in exchange for practical performance. The most common solution is to insert fictitious process noise to increase the steady-state value of the covariance matrix. While these techniques can be successful, they are ad hoc, and can be difficult to implement in a large system.

The method of data forgetting, presented by [1], provides a simple approach that is on firm theoretical ground. This approach represents the weighting of information inversely to its age. In this fashion, the estimate becomes the best fit of the recent data. Optimality is still present in the context of this time weighting. This approach increases the stability of the filter.

In practice, data forgetting presents a single tuning parameter that can be used to adjust filter responsiveness. The following discussion will show that data forgetting can be implemented using a very simple modification to the Kalman covariance propagation. Forgetting can be removed entirely by setting the forgetting rate to zero. In this regard, there is little reason not to include data forgetting in this or any Kalman filter.

The method for imposing data forgetting is most intuitive when viewed in the context of a dual problem for the Kalman filter. The Kalman filter can be shown to also minimize the following cost function [1]:

$$V = \frac{1}{2}\left(x_o - \hat{x}_o\right)^T P_o^{-1}\left(x_o - \hat{x}_o\right) + \frac{1}{2}\int_{t_o}^{t} n_x^T Q^{-1}(\tau) n_x d\tau + \frac{1}{2}\sum_t \left(z - H\hat{x}\right)^T R^{-1}(t)\left(z - H\hat{x}\right) \quad (2\text{-}209)$$

The first term represents the cost of an initial condition error. The second term integrates the cost of process noise. The third term sums the cost of measurement error. In this context, scalar exponential time weighting can be added to this cost function as follows

$$V = \frac{1}{2}e^{2\lambda t_o}\left(x_o - \hat{x}_o\right)^T P_o^{-1}\left(x_o - \hat{x}_o\right) + \frac{1}{2}\int_{t_o}^{t} e^{2\lambda t} n_x^T Q^{-1}(\tau) n_x d\tau + \frac{1}{2}\sum_t e^{2\lambda t}\left(z - H\hat{x}\right)^T R^{-1}(t)\left(z - H\hat{x}\right) \quad (2\text{-}210)$$

Without loss of generality, we can assume that $t_o = 0$, so that the first term is unchanged. This new cost function can be brought to the original form using the substitutions

$$Q_\lambda^{-1}(t) = e^{2\lambda t} Q^{-1}(t) \quad (2\text{-}211)$$

$$R_\lambda^{-1}(t) = e^{2\lambda t} R^{-1}(t)\,, \quad (2\text{-}212)$$

or, alternatively

$$Q_\lambda(t) = e^{-2\lambda t} Q(t) \quad (2\text{-}213)$$

$$R_\lambda(t) = e^{-2\lambda t} R(t)\,. \quad (2\text{-}214)$$

In other words, the optimal solution to the problem with data weighting is identical to the ordinary Kalman filter with modified noise covariance. In particular, both the process noise and measurement noise are reduced exponentially over time. This maintains the balance between measurement and process information, but gives new information more credibility.

The modified covariance can be substituted into the Kalman filter equations to provide a modified formula for filter propagation and measurement updates. However, this result is not suitable for computation, because the noise covariance becomes very small as time increases.

Consider instead the compensated covariance

$$P_\lambda(t) = e^{2\lambda t} P(t). \tag{2-215}$$

Then

$$\dot{P}_\lambda = 2\lambda e^{2\lambda t} P + e^{2\lambda t}\dot{P} = e^{2\lambda t}\left(2\lambda P + AP + PA^T + FQ_\lambda F^T\right). \tag{2-216}$$

The λ's can be split and, since $e^{2\lambda t} Q_\lambda = Q$, this equation can be written as:

$$\dot{P}_\lambda = \left(A + \lambda I\right)P_\lambda + P_\lambda\left(A + \lambda I\right)^T + FQF^T \tag{2-217}$$

The gain can be written in terms of this compensated covariance by substituting $P = e^{-2\lambda t} P_\lambda$ into

$$K = P_- H^T\left(HP_- H^T + LR_\lambda L^T\right)^{-1}, \tag{2-218}$$

which is conveniently unchanged:

$$K = e^{-2\lambda t} P_{\lambda-} H^T\left(He^{-2\lambda t} P_{\lambda-} H^T + Le^{-2\lambda t} RL^T\right)^{-1} = P_{\lambda-} H^T\left(HP_{\lambda-} H^T + LRL^T\right)^{-1}. \tag{2-219}$$

The measurement covariance update

$$P_+ = \left(I - KH\right)P_-, \tag{2-220}$$

is also unchanged:

$$P_{\lambda+} = \left(I - KH\right)P_{\lambda-}. \tag{2-221}$$

In summary, the concept of data forgetting can be employed to the Kalman filter by simply substituting

$$A_\lambda = A + \lambda I \tag{2-222}$$

in the covariance propagation equation. This causes the estimated covariance to grow more rapidly, reaching larger steady state values. This, in turn, creates larger gains and a faster filter response.

Note that the modified covariance P_λ is no longer an estimate of the error covariance. It has been increase exponentially to produce the effect of forgetting. However, this is the covariance that is used to calculate the gain in the Kalman filter. As such, this modified covariance must remain in a region where the model is accurate. When covariance studies are performed using data forgetting, the result will be assessed in this context.

2.2.4 Convergence

The topic of convergence can be approached from several viewpoints. The most common techniques are observability and covariance analysis. Further nonlinear discussion of the extended Kalman filter is also appropriate.

2.2.4.1 Observability

As the optimal linear observer, the Kalman filter will converge for a linear plant if there exists any linear observer that converges. Otherwise, the Kalman filter would not be optimal. A linear observer will converge if the system is observable. It is therefore appropriate to explore the meaning of observability in the context of the continuous-discrete Kalman filter.

Note that observability is predominantly a deterministic concept. It reveals what states can be estimated if there is sufficient measurement quality. Covariance analysis, discussed in Section 2.2.4.2 and conducted using simulations in Section 3.3, reveals how accurate the estimate can be. Strictly speaking, a system is either observable or it is not, but we will sometimes talk about increasing observability in the context of covariance and measurement quality. It is instructive in the observability analysis to consider the effect of ignoring small terms, because they may too small to measure.

A deterministic system is observable if the entire state x can be reconstructed from a finite set of measurements [5, p. 207]. We shall focus on the case where the state can be reconstructed using a linear technique, in the interest of showing that the Kalman filter will converge.

The topic of observability has been studied thoroughly by many authors. Observability conditions for LTI systems are derived in many texts including [5]. The observability of an LTV system is more elusive and can be difficult to determine. The discussion here has been derived independently and contains two distinguishing characteristics. First, it focuses on the subtleties of a continuous system with discrete measurements. Second, the relation between the LTI observability test and LTV observability is explored, for the purpose of providing a practical method of assessing the observability of an LTV system.

LTI Systems

Given the LTI continuous system with discrete measurements

$$\dot{x} = Ax \tag{2-223}$$
$$z_k = Hx_k \tag{2-224}$$

the solution can be expressed as

$$x = e^{At}x_o\,. \tag{2-225}$$

The measurements at intervals T are related to the initial state as follows:

$$z = \begin{pmatrix} z_o \\ z_1 \\ \vdots \\ z_{n-1} \end{pmatrix} = H \begin{pmatrix} x_o \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix} = \begin{pmatrix} H \\ He^{AT} \\ \vdots \\ He^{(n-1)AT} \end{pmatrix} x_o = Mx_o \tag{2-226}$$

Note that equations beyond n - 1 are not linearly independent so they provide no new information. If this matrix M is of rank n, than n independent rows of M and z can be found and solved using

$$x_o = M_n^{-1} z_n\,, \tag{2-227}$$

and the state at any time can then be calculated using (2-225), hence the system is observable. However, it is rather difficult to check the rank of M directly in this form, so we will pursue an alternate expression. The power series expansion

$$e^{At} = \sum_{k=0}^{\infty} \frac{(At)^k}{k!} \tag{2-228}$$

can be rewritten as

$$e^{At} = \sum_{k=0}^{n-1} c_k (At)^k \tag{2-229}$$

where each c_k is a constant that can be determined. This follows from the fact that A^n can be written in terms of lower powers of A [5, p. 487]. An actual expression for c_k will not be needed.

Using (2-229), we can write M as

$$M = \begin{pmatrix} H \\ H\left(c_0 I + c_1 AT + c_2 (AT)^2 + \ldots + c_{n-1}(AT)^{n-1}\right) \\ \vdots \\ H\left(c_0 I + c_1 A(n-1)T + c_2 (A(n-1)T)^2 + \ldots + c_{n-1}(A(n-1)T)^{n-1}\right) \end{pmatrix}. \qquad (2\text{-}230)$$

If the n block rows of M are independent, we can use row reduction to derive a simpler matrix of equal rank. The powers of A can be isolated, by first proceeding downwards to remove the low powers and then upwards to remove the high powers:

$$M \to \begin{pmatrix} H \\ H\left(c_1 AT + c_2 (AT)^2 + \ldots + c_{n-1}(AT)^{n-1}\right) \\ \vdots \\ H\left(c_{n-1}(A(n-1)T)^{n-1}\right) \end{pmatrix} \to \begin{pmatrix} H \\ H(c_1 AT) \\ \vdots \\ H\left(c_{n-1}(A(n-1)T)^{n-1}\right) \end{pmatrix}. \qquad (2\text{-}231)$$

The rows can then be rescaled, yielding

$$M_{LTI} = \begin{pmatrix} H \\ HA \\ \vdots \\ HA^{n-1} \end{pmatrix} \qquad (2\text{-}232)$$

However, for this row reduction to be possible, the original block rows

$$\begin{pmatrix} H \\ He^{AT} \\ \vdots \\ He^{(n-1)AT} \end{pmatrix} \text{ or } \begin{pmatrix} I \\ e^{AT} \\ \vdots \\ e^{(n-1)AT} \end{pmatrix}$$

must be linearly independent. This can be interpreted in the context of sampling. If certain dynamics of A are periodic at exactly the measurement frequency, they will be hidden from the output. While aliasing (which is acceptable) would be quite common, an exact frequency match is unlikely in practical applications. This analysis gives the guidance that the sampling frequency should be chosen to avoid equality with any known system vibration frequencies.

For measurements at aperiodic times, the same results can be achieved, although there are new possibilities for dynamics hidden from the output.

LTV Systems

For a Linear Time-Varying system

$$\dot{x} = A(t)x \tag{2-233}$$

$$z_k = H_k x(t_k) \tag{2-234}$$

we can write similar measurement equations

$$z = \begin{pmatrix} z_o \\ z_1 \\ \vdots \\ z_\infty \end{pmatrix} = \begin{pmatrix} H_o \\ H_1\Phi(t_1,t_o) \\ \vdots \\ H_\infty\Phi(t_\infty,t_o) \end{pmatrix} x_o = Mx_o \tag{2-235}$$

Note that each measurement now has its own *H* matrix and that the matrix exponential e^{At} was replaced with state transition matrix $\Phi(t_1,t_o)$. We are no longer limited to *n*-1 equations.

Once again, if we can find *n* independent rows, we can invert and solve for the initial state and thereby know the state for all time:

$$x(t) = \Phi(t,t_o)x_o \tag{2-236}$$

We would like to simplify the expression for *M* as we did for the time-invariant case. However, the LTV observability problem is much more difficult because the state transition matrix is difficult to evaluate.

It is natural to try to relate the LTV observability to the LTI observability. We will proceed to explore the relationship between LTV observability matrix *M* and the instantaneous LTI observability test matrix $M_{LTI}(t)$.

Let us first consider the following system as an example:

$$A = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{2-237}$$

$$H = \begin{bmatrix} 1 & \sin t \end{bmatrix}. \tag{2-238}$$

The time-invariant observability test matrix is always rank 1:

$$M_{LTI}(t) = \begin{bmatrix} 1 & \sin t \\ 0 & 0 \end{bmatrix}. \tag{2-239}$$

Since $\Phi = e^{At} = I$, the first few rows of the time-varying observability are

$$M = \begin{bmatrix} 1 & \sin t_o \\ 1 & \sin t_1 \\ 1 & \sin t_2 \end{bmatrix}. \tag{2-240}$$

This is rank 2, provided that the measurements do not occur at periods of π, so the LTV system is observable. It is common, as in this example, for the instantaneous LTI observability test matrix $M_{LTI}(t)$ to underestimate the observability of an LTV system.

This can be formalized by viewing the LTV system as a series of piecewise LTI systems. Consider a system that is LTI for a short window of time during which *r* measurements arrive at period T. As the LTI time window can be made arbitrarily small, this is not a restrictive assumption. Then for the samples $k < r$, the state can be expressed using the state transition matrix:

$$x(t+kT) = \Phi(t+kT, t_o)x_o = \Phi(t+kT, t)\Phi(t, t_o)x_o, \tag{2-241}$$

which has been separated into the LTV and LTI transitions. Using the LTI approximation, we have

$$x(t+kT) \approx e^{A(t)kT}\Phi(t, t_o)x_o \tag{2-242}$$

Since $H(t)$ is approximately constant during this time period, the measurement is

$$z_k = z(t+kT) = H(t)e^{A(t)kT}\Phi(t, t_o)x_o \tag{2-243}$$

A portion of the observability matrix can be formed using these *r* measurements of the LTI time window:

$$z = \begin{pmatrix} z_o \\ z_1 \\ \vdots \\ z_r \end{pmatrix} = \begin{pmatrix} H(t) \\ H_1(t)e^{A(t)T}) \\ \vdots \\ H_r(t)e^{A(t)rT} \end{pmatrix} \Phi(t, t_o)x_o. \tag{2-244}$$

The LTV state transition matrix is full rank because

$$\Phi^{-1}(t_1, t_o) = \Phi(t_o, t_1). \tag{2-245}$$

The LTI portion of (2-244) contains the first *r* rows that were used in LTI observability analysis. If $r \geq n$, than this rank can be checked using the instantaneous LTI observability test matrix:

$$\text{rank}\begin{pmatrix} H(t) \\ H_1(t)e^{A(t)T} \\ \vdots \\ H_r(t)e^{A(t)rT} \end{pmatrix} = \text{rank}\left(M_{LTI}(t)\right). \tag{2-246}$$

In other words, if *n* measurements arrive during the LTI time window, than the LTI observability test matrix can be used to predict the LTV observability. This will underestimate observability because it only considers the measurements from one time window.

The full LTV observability matrix can be assembled using sets of measurements from successive time windows:

$$z(t_1 + kT) = H(t_1)e^{A(t_1)kT}\Phi(t_1, t_o)x_o \tag{2-247}$$

$$z(t_2 + kT) = H(t_2)e^{A(t_2)kT}\Phi(t_2, t_o)x_o \tag{2-248}$$

$$z(t_3 + kT) = H(t_3)e^{A(t_1)kT}\Phi(t_3, t_o)x_o \tag{2-249}$$

The resulting test matrix can be written as

$$M_{LTV} = \begin{bmatrix} M_{LTI}(t_1)\Phi(t_1, t_o) \\ M_{LTI}(t_2)\Phi(t_2, t_o) \\ M_{LTI}(t_3)\Phi(t_3, t_o) \\ \vdots \end{bmatrix}. \tag{2-250}$$

Alternatively, the first transition can be removed

$$M_{LTV} = \begin{bmatrix} M_{LTI}(t_1) \\ M_{LTI}\Phi(t_2, t_1) \\ M_{LTI}\Phi(t_3, t_1) \\ \vdots \end{bmatrix}. \tag{2-251}$$

This expression provides a method to test the observability of an LTV system using the LTI observability test matrix at different time intervals. It is valid only for $r \geq n$, when the LTI approximate is valid for *n* measurements. This expression is most useful during special cases when the state transition matrix can be easily calculated.

In the event that the LTI approximation is only valid for short time periods, resulting in $r < n$, the row reduction leading to M_{LTI} as given by (2-231) will not be possible and (2-251) cannot be used. This is because there are fewer rows than there are powers of *A*.

To the extent that

$$e^{AT} \approx \sum_{i=0}^{r-1} \frac{(AT)^i}{i!}, \tag{2-252}$$

the first r rows of $M_{LTI}(t)$ can be used in (2-251). This approximation uses the first r terms of the power series expansion and is valid when the sample rate is fast compared to the dynamics. If this is not the case, then matrix exponential in (2-244) must be evaluated from the full series.

For systems with rapid time variance, it may be preferable to compute the state transition matrix and evaluate (2-235) directly. This can be done by calculating

$$\Phi(t, t_o) = \exp\left(\int_{t_o}^{t} A(\tau) d\tau \right), \tag{2-253}$$

using numerical integration and the series expansion of the exponential.

Nonlinear Systems

In the case of a nonlinear system of the form

$$\dot{x} = f(x,t) \tag{2-254}$$

$$z_k = h(x_k, t), \tag{2-255}$$

there is no simple method to check observability, although the notion still has meaning. If the system is well approximated by a linear system that is observable, then an approximate solution can be estimated. The process of linearization typically removes information, so there are many cases where an observable nonlinear system has an unobservable linearization. Consider the measurement $z = x^3$, which provides no information when linearized about $x = 0$. It also may be possible for the opposite to be true if limit cycles appear at the sampling frequency.

In the context of the Extended Kalman Filter, observability of the linearized system is required. This is because the filter uses a linearization, and is therefore blind to nonlinear effects. For this reason, a nonlinear observability test is not required.

Stochastic Stability

Rigorous stability analysis for the continuous Kalman filter has been performed in [12] and [2]. Conditions can be established that guarantee the exact convergence of the estimate in the presence of noise. These results are readily applicable to the continuous-discrete filter.

As in the deterministic observer, observability is required so that errors can be detected. Additionally, it is required that the filter equations remain finite. This requires bounded state matrices and process and measurement noise. In the continuous Kalman filter, it is also required that the measurement noise is nonzero so that the Kalman gain remains finite.

The third requirement is that the state is controllable through the process noise. In other words, the process noise must be able to influence all of the state variables. This requirement prevents the covariance from becoming too small. Controllability can be tested using methods similar to testing observability.

Under these conditions, the Kalman filter is asymptotically stable. With the stronger conditions of uniform observability and uniform controllability, the Kalman filter is exponentially stable. Tests for uniform observability and controllability are beyond the scope of this thesis.

Detectability

It is noted in [2] that the Kalman filter will perform successfully in the presence of unobservable states if these states are all stable. Formally, this requirement is termed *detectability*. With this condition, the unobservable modes will decay, and a linear observer converges correctly.

This subtlety can provide two benefits. First, states that may be observable can be included in the filter, so long as they are stable. For example, states that are only observable during certain time periods or conditions can still be used. This approach can lead to the higher performance.

Second, unobservable states can be distinguished by their probabilities, since the Kalman filter is still choosing the optimal estimate. Unobservable states may be necessary for noise modeling. This can allow the error covariance to propagate correctly even though these states cannot be estimated precisely. However, unobservable states should usually be avoided because additional states are a large computational burden and the exact probabilities are usually unknown.

Unstable or marginally stable unobservable modes will cause great problems in a Kalman filter. Such modes will cause the covariance matrix to grow without bounds, leading to numerical problems. Also, poorly observable states with huge variance can still “draw the blame” for measurement errors and compromise the estimates of other states.

2.2.4.2 Covariance Analysis

The Kalman filter provides its own method for studying and monitoring filter convergence in the full stochastic sense: the covariance matrix. The diagonal elements, or the principal values, give a measure of the estimation confidence. Convergence of the Kalman filter requires that the covariance matrix is bounded. The actual values of the covariance matrix give the predicted accuracy of the filter.

The covariance matrix can be propagated offline to predict performance. It can also be monitored in real time during the actual application. In contrast to observability, covariance analysis is simpler to conduct and also produces a characterization of accuracy.

An important point is that covariance analysis assumes that the model is correct. The fact that the filter thinks it has converged does not guarantee convergence. It can actually be a practical problem if the covariance becomes too small, because the filter will then ignore future measurements and become unresponsive.

2.2.4.3 The Effect of Modeling Errors

The effect of modeling errors is presented in [7, pp. 250-254] and [13, pp. 205-218] for continuous-time and discrete-time Kalman filters. The development here extends those results to apply to the mixed continuous-discrete Kalman filter.

Suppose that the true system dynamics are given by the time-varying matrices A, F, H, D, and L, as defined earlier (2-102, 2-103), but that the Kalman filter uses a model with $\hat{A}$, $\hat{F}$, $\hat{H}$, $\hat{D}$, and $\hat{L}$. Define the modeling errors by the differences $\delta A = A - \hat{A}$ and so forth. We wish to express the propagation of the actual estimate error in terms of these modeling differences.

The filter propagation is now

$$\dot{\hat{x}} = \hat{A}\hat{x} + u \tag{2-256}$$

with error dynamics

$$\dot{e} = \dot{\hat{x}} - \dot{x} = \hat{A}\hat{x} + u - Ax - u - Fw = \hat{A}e - \delta A x - Fw . \tag{2-257}$$

This can be written using an augmentation of the state vector, with the actual dynamics in the second row:

$$\begin{pmatrix} \dot{e} \\ \dot{x} \end{pmatrix} = \begin{bmatrix} \hat{A} & -\delta A \\ 0 & A \end{bmatrix} \begin{pmatrix} e \\ x \end{pmatrix} + \begin{bmatrix} -F \\ F \end{bmatrix} w . \tag{2-258}$$

The covariance propagation expression (2-135) still holds for these augmented matrices:

$$\dot{P}' = A'P' + P'A'^T + F'QF'^T, \tag{2-259}$$

where

$$A' = \begin{bmatrix} \hat{A} & -\delta A \\ 0 & A \end{bmatrix} \tag{2-260}$$

$$F' = \begin{bmatrix} F \\ -F \end{bmatrix} \tag{2-261}$$

$$P' = E\begin{bmatrix} ee^T & ex^T \\ xe^T & xx^T \end{bmatrix}. \tag{2-262}$$

Measurement updates can be treated similarly. The observer update is now

$$\hat{x}_+ = \hat{x}_- + K\left(Hx_- + Ln - \hat{H}\hat{x}_-\right) \tag{2-263}$$

which gives the error update

$$e_+ = \left(I - K\hat{H}\right)e_- + K\delta Hx + KLn. \tag{2-264}$$

This again can be written using the augmented state (the second equation is trivial):

$$\begin{pmatrix} e_+ \\ x_+ \end{pmatrix} = \begin{bmatrix} I - K\hat{H} & K\delta H \\ 0 & I \end{bmatrix}\begin{pmatrix} e_- \\ x_- \end{pmatrix} + \begin{bmatrix} KL \\ 0 \end{bmatrix}n. \tag{2-265}$$

Multiplying by the transpose and taking the expectation yields

$$P'_+ = E\begin{bmatrix} e_+e_+^T & e_+x_+^T \\ x_+e_+^T & x_+x_+^T \end{bmatrix} = \begin{bmatrix} I - K\hat{H} & K\delta H \\ 0 & I \end{bmatrix}P'\begin{bmatrix} I - K\hat{H} & K\delta H \\ 0 & I \end{bmatrix}^T + \begin{bmatrix} KL \\ 0 \end{bmatrix}R\begin{bmatrix} KL \\ 0 \end{bmatrix}^T, \tag{2-266}$$

since the measurement noise n is independent of e_- and x_-. The gain K in these equations is generated using the incorrect model with the Kalman Filter equations given earlier:

$$\dot{P} = \hat{A}P + P\hat{A}^T + \hat{F}Q\hat{F}^T \tag{2-267}$$

$$K = P_-\hat{H}^T\left(\hat{H}P_-\hat{H}^T + \hat{L}R\hat{L}^T\right)^{-1} \tag{2-268}$$

$$P_+ = \left(I - K\hat{H}\right)P_- \tag{2-269}$$

Note that δA and δH appear directly in the equations for P' while modeling errors is F, D, and L, as well as the noise characteristics Q and R, led to a suboptimal Kalman gain K.

With these results, we have equations for propagating P' between and during measurements. While the entire P' matrix must be propagated, the upper left partition of P' is of interest. This is the actual estimation error covariance using the incorrect model, and should be compared to the Kalman Filter's covariance matrix P.

The above procedure is primarily useful in numerical simulation. Calculation of the P' matrix at different levels of model accuracy will illustrate the modeling sensitivity. This procedure can show the statistical effect of a modeling error on estimation performance. It can also be used to establish conditions for the stability of P', which governs the actual convergence of the Kalman Filter.

2.2.4.4 Nonlinear Convergence

Convergence of the Extended Kalman Filter is a difficult issue that is frequently avoided. Because the Kalman filter is only optimal for LTV systems, no general statement about stability can be made.

Since the EKF uses a linearization of the state equations about the estimated state, there is a potential for large modeling errors. If there is an estimation error, the linearization will be taken about the wrong point. Additionally, the process of linearizing produces errors which may be large in certain systems.

The first case of incorrect linearization can be treated using the previous analysis of modeling errors. Here the real system can still be well represented by a linear system, and matrices such as δA can be defined. The relationship between the nonlinear state estimation errors and the errors in the state matrices must be analyzed. In principle, a bounding region of convergence can be found, perhaps through the use of numerical simulation.

In the second case, the true system cannot be represented by a linear system. As a result, the linear modeling error analysis is not applicable. For these systems, the use of an Extended Kalman Filter is questionable and performance issues are likely.

2.2.4.5 Convergence with Data Forgetting

Stability of the Kalman filter with data forgetting is discussed in [1]. It is found that data forgetting increases the stability of the filter uniformly by the forgetting factor λ. This means that an asymptotically stable Kalman filter will become exponential stable at rate λ, and a Kalman filter that is already exponential stable will converge faster by rate λ.

The argument can be presented as follows. The Kalman filter with data forgetting has the identical gain as the ordinary Kalman filter with

$$A_\lambda = A + \lambda I \,. \tag{2-270}$$

If this gain were used to estimate a system that naturally contained A_λ, the continuous-time error dynamics using (2-129) would be

$$\dot{e}_\lambda = A_\lambda e_\lambda - Fw\,. \tag{2-271}$$

Using (2-270), the error dynamics using forgetting on the actual system A are then

$$\dot{e} = Ae - Fw = \left(A_\lambda - \lambda I\right)e - Fw\,. \tag{2-272}$$

If the Kalman filter for the A_λ system converges, then this error e is bounded by exponential convergence at rate λ. Once again, this is because the discrete-time measurement updates have the same gain in both filters. The matrices A and A_λ will provide the same conditions of observability, finiteness, and controllability that were needed to establish the convergence of this error.

Note that data forgetting added stability to the error while adding instability to the state dynamics. The covariance is artificially increased and gains are augmented. This brings a subtle caveat with dectectable systems. It is possible for stable modes to become unstable when data forgetting is employed. Since the preceding argument required the convergence under A_λ, it is required that unobservable modes decay faster than the forgetting rate.

2.2.5 Implementation Techniques

Several additional topics are important for the implementation of the Kalman filter. These next sections address common practical issues from a theoretical point of view.

2.2.5.1 Multirate Measurements and Dropouts

It is common that measurements arrive at different rates, or that sensors experience dropout periods. In these circumstances, partial measurement updates are required. Such processing is directly possible with two alternative techniques.

The most straightforward method is to zero the rows of the measurement matrix H that correspond to the absent measurements. In this fashion, all other measurement can still be processed normally and the filter framework is unchanged.

A second technique is to completely remove the corresponding rows of H and compress the measurement vector z. This has the potential of reducing the computational load, especially in the matrix inversion (2-158). This technique may be preferable if it can be guaranteed that certain measurements never arrive at the same time. For example, there will always be some GPS satellites that are behind the Earth. However, this second technique quickly introduces new programming difficulties and may require dynamic memory allocation.

In the simulation and hardware implementation for this thesis, the first method is used to handle measurement dropouts.

2.2.5.2 Modeling of Colored Noise

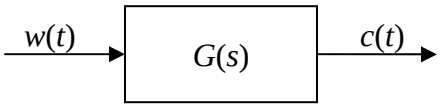

Figure 2-3: Model of Colored Noise

Although the Kalman filter requires the assumption that noise is white (time-uncorrelated), a straight forward procedure exists for accommodating colored noise. As shown in Figure 2-3, colored noise, denoted $c(t)$, can be represented as the output of a linear system that is driven by white noise $w(t)$. The linear transfer function $G(s)$ must be added to the process dynamics using additional state variables.

The transfer function $G(s)$ can be determined using a variety of methods. In some cases, the structure may be known from the physics of the noise process. More often, the noise characteristics are known only from experimental data. In this case, the autocorrelation and power spectral density of the noise can be estimated from the data. Several techniques exist for fitting a transfer function to such data. Most require selecting a model set, such as an all-

pole filter, and then finding a Least Squares solution. It is usually easier to fit a discrete-time transfer function and then convert the result to a continuous-time equivalent.

It is convenient to use a first order continuous-time model of the form

$$\dot{c}(t) = -rc(t) + w(t). \tag{2-273}$$

Note that this filter has a gain of r. The variance of the colored noise can be expressed using the continuous covariance update of the Kalman filter

$$\dot{P} = AP + PA^T + FQF^T. \tag{2-274}$$

For a simple scalar equation, with $A = -r$ and $F = 1$, this reduces to

$$\dot{\sigma}_c^2 = -2r\sigma_c^2 + \sigma_w^2, \tag{2-275}$$

and has a steady state value of

$$\sigma_c^2 = \frac{\sigma_w^2}{2r}. \tag{2-276}$$

This result is counterintuitive because the filter did not have unity gain. If the gain is removed from the filter, (2-274) can be evaluated with $F = r$ and the result becomes $\sigma_c^2 = r\sigma_w^2/2$. While this result is useful conceptually, (2-276) is more convenient.

The autocorrelation of the colored noise, defined by $E\{c(t)c(t-T)\}$ for $T \geq 0$, can be calculated using the convolution solution

$$c(t) = \int_{t_o}^{t} e^{-r(t-\tau)} w(\tau) d\tau. \tag{2-277}$$

It follows that

$$c(t)c(t-T) = \int_{t_o}^{t} e^{-r(t-\tau_1)} w(\tau_1) d\tau_1 \int_{t_o}^{t-T} e^{-r(t-\tau_2)} w(\tau_2) d\tau_2. \tag{2-278}$$

The distinct integration variables τ_1 and τ_2 have been used so that the product can be written as a double integral:

$$c(t)c(t-T) = \int_{t_o}^{t-T} \int_{t_o}^{t} e^{-r(t-\tau_1)} w(\tau_1) e^{-r(t-\tau_2)} w(\tau_2) d\tau_1 d\tau_2. \tag{2-279}$$

Note that for white noise

$$E\{w(\tau_1)w(\tau_2)\} = \sigma_w^2 \delta_c(\tau_1 - \tau_2), \tag{2-280}$$

where $\delta_c(t)$ is the Dirac delta function. Therefore, taking the expectation and combining the exponentials produces

$$E\{c(t)c(t-T)\} = \int_{t_o}^{t-T}\int_{t_o}^{t} e^{-r(2t-\tau_1-\tau_2)}\sigma_w^2\delta(\tau_1-\tau_2)d\tau_1 d\tau_2 . \tag{2-281}$$

Because of the sifting property of the Dirac delta function, the inner integral is nonzero only when $\tau_1 = \tau_2$ and we have

$$E\{c(t)c(t-T)\} = \int_{t_o}^{t-T} e^{-2r(t-\tau_2)}\sigma_w^2 d\tau_2 . \tag{2-282}$$

This can be evaluated without difficulty, and the solution is

$$E\{c(t)c(t-T)\} = \frac{\sigma_w^2}{2r}\left(e^{-2rT} - e^{-2r(t-t_o)}\right). \tag{2-283}$$

The second term inside the parentheses represents the transient, and vanishes in steady state. The autocorrelation function is then dependent on T alone, so this is a Wide Sense Stationary process. For $T = 0$, this result matches the expression for steady state variance given by (2-276). As T increases, the autocorrelation approaches zero exponentially.

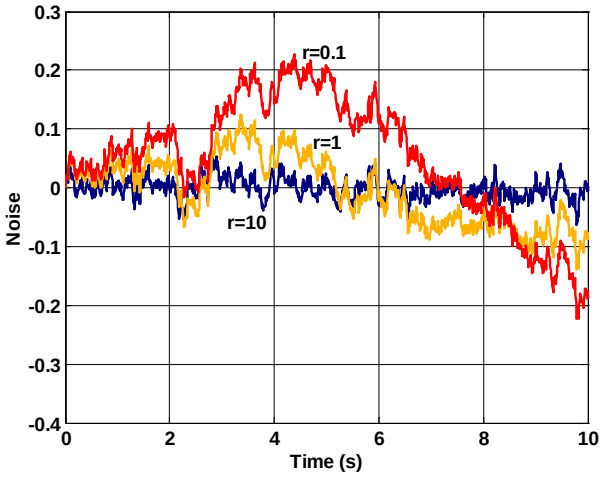

Figure 2-4: Colored Noise with Varying Filter Rate

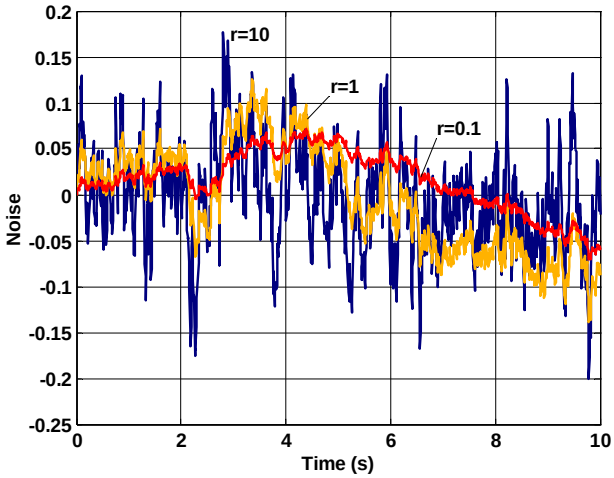

Figure 2-5: Colored Noise with Equal Variance

It is instructive to look at the time behavior of this colored noise. Figure 2-4 shows colored noise at varying filter rates with constant white noise variance. This was generated using a discrete approximation at 100 Hz with a pseudorandom number generator. A slower rate corresponds to a larger time constant, which smoothes the noise but also increasing the gain. It is more instructive to hold the variance of the colored noise constant, by changing the white noise variance and filter rate in proportion. The result is plotted in Figure 2-5. Small filter rates produce highly correlated noise that appears smoother and varies more slowly.

2.2.5.3 Numerical Techniques

Several computational techniques are relevant for the practical implementation of the Kalman filter. These methods can improve performance under imperfect arithmetic and real-time processing.

A host of techniques are available for improving the numerical robustness of the solution. It is critical that the covariance matrix remains symmetric positive definite. Otherwise, it is not a meaningful covariance and the gains and estimate updates will be erroneous. The matrix inversion in the measurement update can also fail.

Maintaining covariance symmetry is a simple matter. The solution can be "symmetricized" periodically using

$$P_+ = \frac{1}{2}\left(P_- + P_-^T\right). \tag{2-284}$$

Another approach is to implement the filter using entirely symmetric equations. The continuous propagation is already symmetric, and the measurement update can be implemented using Joeseph's Stabilized form (2-160).

Maintaining positive symmetric definiteness is more difficult. Several algorithms factor the covariance into a form that enforces this, such as the square root Cholesky decomposition [21, p. 15]:

$$P = LDL^T = \left(L\sqrt{D}\right)\left(L\sqrt{D}\right)^T, \tag{2-285}$$

Here D is a diagonal matrix that has positive entries when P is symmetric positive definite and L is another matrix. The Kalman filter can then be written in terms of these factors. A discussion of these algorithms, along with charts of their memory and computation implications, can be found in [8, p. 244]. These algorithms generally require more computation because the factored equations are more complicated.

The matrix inversion in the measurement update produces a large computational burden for large systems. This inversion can be eliminated by processing the measurements one at a time [8, pp. 232-233]. The technique requires the measurement covariance LRL^T to be diagonal, which we will name R_{diag}.

The one-at-a-time update routine loops through all measurements, with the update for the i^{th} measurement z_i given by:

$$K = \frac{P_{\_} H_i^T}{H_i P_{\_} H_i^T + R_{ii}} \tag{2-286}$$

$$x_+ = x_{\_} + K\left(z_i - H_i x\right) \tag{2-287}$$

$$P_+ = P_{\_} - K H_i P_{\_}, \tag{2-288}$$

where H_i is the i^{th} row of H and R_{ii} is the element in the i^{th} row and i^{th} column of R_{diag}. Note that the state and covariance update must happen in each iteration, so that the updated covariance is used in successive gain calculations.

If the measurement covariance is not diagonal, it can always be decoupled using a change of measurement variables. For example, the GPS noise may have principal directions in the NED coordinate from, instead of ECF. An alternative is to process measurements "3 at a time", or "*n* at a time" if blocks of size *n* are coupled.

Simulation in software does not require most of these techniques. In my simulations, I have used double precision, floating-point arithmetic because there was an abundance of memory. However, making *P* symmetric periodically is highly beneficial for alleviating numerical instabilities.

Real-time hardware implementation is more demanding. Processing speed is critical, and can be greatly improved using one-at-a-time measurement processing as well as several additional programming techniques that are discussed in Section 4.3. I did not employ square root filtering in the real-time implementation, but this could be beneficial because single precision arithmetic is used on the processor.

2.2.5.4 Filter Health Monitoring

In a high-order Kalman filter, it is important to have a simple indicator of filter performance. This can be used in real time to detect anomalous conditions, or during simulations to detect modeling errors.

A measure of filter health can be derived from the information matrix

$$Y = \left(H P_{\_} H^T + L R L^T\right)^{-1}, \tag{2-289}$$

which appears in the Kalman gain equation. The size of *Y* can be characterized by the maximum singular value. When *Y* is small, the state is well known and the new measurements provide little information. When *Y* is large, the state is poorly known and the new measurements provide a lot of information. The Kalman gain is then proportional to the information matrix *Y*.

Comparison of the information matrix to the measurement error $\tilde{z}$ gives the likelihood that this measurement would have occurred. For example, a large measurement error is unlikely when the information matrix is small. For a Gaussian distribution, this likelihood function can be calculated from the probability distribution function, producing

$$l(\tilde{z}) = \exp\left(-\frac{1}{2} \tilde{z}^T Y \tilde{z} \right). \tag{2-290}$$

The result is simplified by using the negative log likelihood, and normalizing by the number of measurements *m*:

$$\chi^2 = \frac{\tilde{z}^T Y \tilde{z}}{m}. \tag{2-291}$$

This is a Chi-squared statistic, which increases when the measurements do not fit the model. This statistic is tabulated in [8, p. 247]. For example, a chi-squared value of 5 has a 2% chance of being a valid occurrence.

In healthy filter operation, the chi-squared value should be small. A measurement glitch will produce a spike in the chi-squared value. A divergent filter will have a chi-squared statistic that increases with time.

With the previous discussions on navigation and the Kalman filter, we are now ready to design this navigation system.

2.3 Application of the Kalman Filter to Navigation

The selection of states and measurements for the Kalman filter is a central issue in designing a navigation system. While most navigation systems use a Kalman filter in some form, there is great flexibility in format. Ultimately, the system performance is greatly impacted by these choices. There are a variety of common architectures and choices of state variables [3, 4, 6, 8, 9, 15, 17], many of which have influenced my design.

There are several fundamental decisions in the architecture that arise in the design of most navigation systems. The first is the treatment of angles and the selection of an attitude representation. The second is the choice of the process dynamics, including the modeling of bias and error states. Finally, there is some flexibility in the choice of measurements, especially when considering that additional system knowledge can be treated as a measurement.

Attitude Representation

An innovative feature of this filter is the combination of quaternions and small angles. This format utilizes the quaternions for singularity-free large angle motion while using small angles for a linear representation of orientation error. In effect, the four quaternion parameters are linearized using three small angles. These small angles are a minimal representation of attitude which is more suitable for the Kalman filter. In addition, small angle rotations are commutative and can be represented using a vector cross product, so the filter equations become reasonably simple.

The use of quaternions directly in the Kalman filter is fairly common [6], and can be successful if the quaternion normality constraint is enforced. To be optimal, special attention must be given to the process noise to ensure that the uncertainty in the quaternion estimate always runs along the quaternion surface. This can be strengthened by treating the normality constraint as an additional measurement. Still, this approach remains undesirable because it adds one extra state variable [8, p. 258]. This creates excess computation, at best, and can lead to observability problems.

It should be stressed that the use of small angles in this filter is no more of an approximation than the linearization required to implement any Extended Kalman Filter. The inaccuracy of attitude linearization is present regardless of whether small angles are used. Consider instead using Euler angles for representation of the orientation error. With the immediate error state offloading scheme that was discussed in Section 2.2.2.3, the estimated orientation error will always be zero. As a result, the Euler error state should be linearized about the point of zero rotation. Evidently, small angles are this linearization. A similar argument applies to rotation vectors. The technique of using small angles directly as the state variables provides a linearization equivalent to the full nonlinear angle treatment, but with greatly simplified equations.

Rigid Body Dynamics as the Process

In this filter, I have incorporated rigid body motion as the process. In this fashion, force and torque are treated as inputs, and the laws of physics are enforced. These inputs are divided into a known part and a random part, as shown in Figure 2-6. The known part passes directly into the filter using the deterministic control input (u). The random part is modeled using additional state variables and is estimated by the filter.

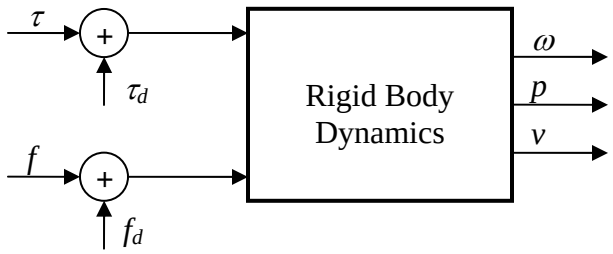

Figure 2-6: Filter Implementation of Rigid Body Dynamics

The popular alternative is to use inertial navigation as the process. In that architecture, the IMU measurements are is treated as process inputs instead of measurements. This approach is detailed in the Appendix. The GPS becomes the sole measurement, and the number of filter measurements and states is reduced.

The use of rigid body dynamics has the potential for increasing response and accuracy in applications where the force and torque are accurately known. The prime example is exoatmospheric flight where the dynamics are extremely predictable. Applications using stabilized platforms or active vibration suppression systems also fit well into this architecture because there is knowledge and measurement of the actuation efforts.

In situations where force and torque are unpredictable, the disturbances are estimated by the filter. Rigid body dynamics can still be beneficial if the force and torque can be modeled adequately. However, this adds complexity and several state variables, and may be inferior in certain applications.

With these process dynamics, the filter may be able to survive an IMU measurement dropout. This would be a difficult situation for any filter, but the propagation of the state using the physics of motion is the most logical solution.

A first order disturbance model has been chosen for general purpose operation. Here, the force and torque disturbances are modeled using exponential correlated noise. As described in Section 2.2.5.2, there is one rate parameter describing the correlation of each dimension of the disturbance.

Loosely-Coupled Architecture

The difference between a tightly-coupled and loosely-coupled implementation is shown in Figure 2-7. In a tightly-coupled architecture, the GPS pseudoranges ρ are the raw measurements, and GPS error states are estimated along with the navigation states in a single Kalman filter shown at the top of the figure. In a loosely-coupled architecture, the GPS position p_{gps} and velocity v_{gps}, which are calculated values, are treated as the raw measurements and are passed to the navigation Kalman filter shown at the bottom of the figure. The GPS solution and compensation occurs in a separate algorithm, which may be implemented using another Kalman filter.

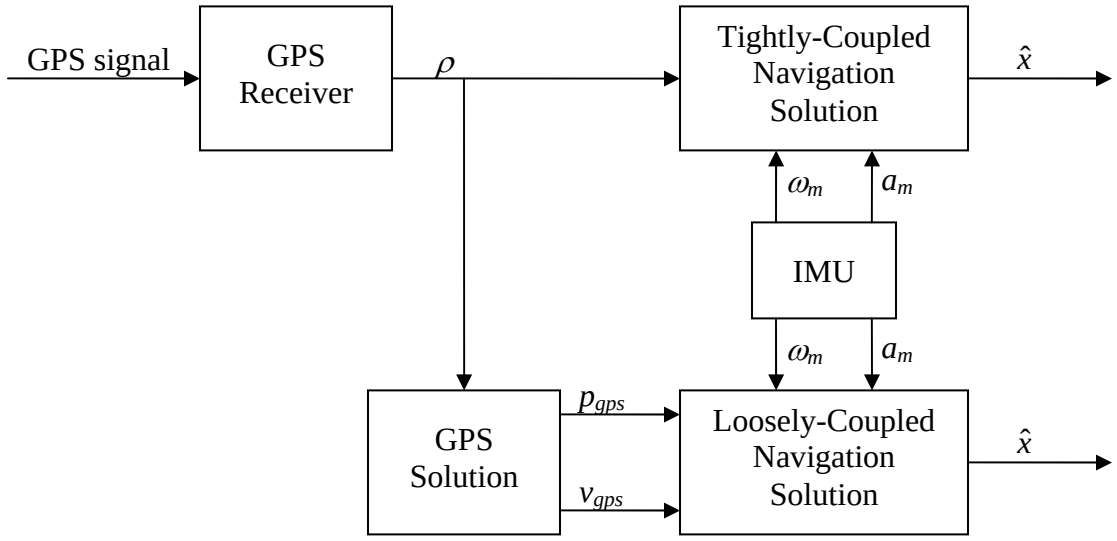

Figure 2-7: Tightly-Coupled and Loosely-Coupled Architectures

The tightly-coupled architecture is superior from a theoretical point of view. An analysis of a decentralized Kalman filter, in which multiple Kalman filters operate on different parts of the state vector, is conducted in [3, pp. 371-377]. The primary result is that decentralized Kalman filters are suboptimal unless full state covariance information is maintained by all filters. That, of course, defeats the purpose of decentralizing.

The loosely-coupled architecture has several practical advantages, which have ultimately led to its selection here. The first is that it has a smaller state vector. Tight-coupling requires at least one state and one measurement for the pseudorange of each GPS satellite, which can dramatically increase the computational burden. The resulting filter may be unfeasible for real hardware, or may require a slow sample rate that could repeal the benefits of theoretical optimality.

Additionally, and most important in a practical sense, the GPS position and velocity solution is an output common to all GPS receivers. In using the loosely-coupled architecture, the user is free to choose from all available commercial and military receivers and can benefit from proprietary algorithms and implementation hardware in these devices. A variety of GPS augmentation services, including Differential GPS, can be readily integrated to improve the

solution quality. The sum of these advantages may produce more accuracy than processing the raw pseudorange measurements.

Finally, some elements of the GPS solution are not suitable for a Kalman filter. For example, ambiguity resolution for phase range measurement requires an algebraic solution. In situations where the dynamics are trivial, a Least Squares solution is usually preferred. It may also be inappropriate to model the pseudorange errors of each satellite as independent variables. For these reasons, it is unrealistic to process all GPS information using a Kalman filter.

With the loosely-coupled implementation, there are two methods of sharing information that can increase performance and limit the suboptimality of decentralized filtering. First, the accuracy of the GPS solution should be given to the navigation Kalman filter as a measurement covariance. This accuracy can be computed from the GPS Dilution of Precision (DOP), which is output by the receiver and usually has different values in the horizontal and vertical directions. If Differential GPS or another aid is available, the measurement covariance should be updated to reflect this increase in accuracy.

Second, the vehicle velocity and acceleration should be passed into the GPS receiver, so that it may compensate for fast vehicle dynamics. Unfortunately, most GPS receivers do not accommodate this. If available, this could drastically reduce the delay in the GPS solution and provide GPS measurements that are truly uncorrelated from process dynamics.

Choice of Measurements

The IMU and GPS are the primary measurement sensors and should naturally be selected as measurements in the Kalman filter. This includes body rates and accelerations from the IMU, and the position and velocity solution from the GPS.

There is additional data that can be interpreted as a filter measurement. First, when the vehicle is Earth stationary (fixed to ground) the rotation rate and velocity are determined. This knowledge can be treated as measurements in the Kalman filter. This effectively allows the filter to perform gyrocompassing and calibrate the IMU any time the vehicle is stationary.

This measurement brings several operating modes into a single Kalman filter. It is common for navigation systems to stop the Kalman filter during gyrocompassing and for Earth stationary operation. For example, the position estimate can be held constant when the vehicle is fixed to Earth. However, stopping and restarting the Kalman filter can produce transients in the estimate with varying accuracy. It is preferable to use a single Kalman filter for initial alignment, fixed-to-Earth operation, and normal flight. It should be noted that a separate initial alignment procedure is always needed when there are large initial errors.

A second measurement can be derived from knowledge of the force disturbance. This could be a real measurement in some systems. It can also be knowledge that the vehicle is in exoatmospheric freefall. This measurement is designed to improve the observability problem that is encountered during freefall (Section 2.3.3).

Measurement Error Modeling

Since the Kalman filter assumes white noise, it is often necessary to add states to model time-correlated errors in the measurements. This can be done with arbitrary accuracy using the techniques of Section 2.2.5.2. High-order noise models, however, can greatly increase the computation time. Ultimately, we are unconcerned with the accuracy of the noise estimate, provided that the Kalman filter produces accurate estimates of the navigation states.

For the IMU, gyroscope and accelerometer errors arise from a variety of sources. For the most part, these errors can be sufficiently modeled using a bias and a scale factor. Additional effects exist [8], such as the gyroscope error that arises from structural deflection during periods of acceleration.

In this filter implementation, IMU errors are modeled with biases alone. In this fashion, bias states are used to estimate the error caused by both biases and scale factors. The benefit of using scale factors is not realized until the trajectory contains a rich variety of dynamics. The cost of IMU scale factors is an additional 6-18 states, depending on whether cross coupling states are included. When modeling using bias states alone, the noise characteristic of the bias must include the effect of a scaling error.

The GPS position solution contains many sources of error, ranging from inaccuracies in system clocks to atmospheric disturbances that alter the transmission time. These inaccuracies would be modeled in a tightly-coupled architecture. For our loosely-coupled architecture, the modeling of a GPS measurement bias was considered. However, there is no information to observe this bias; an error in GPS position is virtual undetectable without a redundant position measurement. For this reason, the GPS noise was assumed to be white.

The fixed-to-Earth measurement requires a noise model, because the vehicle will not be perfectly stationary. In effect, the vibration of the vehicle becomes the measurement noise. In this filter, the vibration is assumed to be white. However, this is not accurate when considering structural resonances, such as a rocket on the launch rail. Higher performance could possibly be achieved by using higher-order noise modeling.

Summary of Filter Architecture

The choice of states is shown in the Table 2-1. Because of the use of quaternions and small angles, the full nonlinear state vector has one more element that the linear state vector. The state vector includes attitude (quaternions and small angles), body rates, torque disturbance, position, velocity, force disturbance, and biases for the gyroscopes, accelerometers, and gravity model.

Measurements are shown in Table 2-2. Measurements include the gyroscopes, accelerometers, rate and velocity when Earth stationary, torque and force disturbance, and GPS position and velocity.

Control inputs are shown in Table 2-3. As discussed, the torque and force are inputs to the model. The specific values (per unit mass) have been used so that inertia properties are not

needed here. An additional known parameter is the angle of the Earth, which is treated as a control input.

Table 2-1: Filter States

Full Index	Error Index	Size	Symbol	Description	Units
1:4		4	q	Quaternions	-
	1:3	3	α	Small angle misalignment	rad
5:7	4:6	3	ω	Body rates	rad/s
8:10	7:9	3	τ_d	Specific torque disturbance	rad/s^2
11:13	10:12	3	p	ECI position	m
14:16	13:15	3	v	ECI velocity	m/s
17:19	16:18	3	f_d	Specific force disturbance	m/s^2
20:22	19:21	3	b_ω	Gyro bias	rad/s
23:25	22:24	3	b_a	Accelerometer bias	m/s^2
26:28	25:27	3	b_q	Gravity model bias	m/s^2
29:31	28:30	3	b_t	GPS time delays	s

Table 2-2: Filter Measurements

Index	Size	Symbol	Description	Units
1:3	3	ω_m	Gyro rates	rad/s
4:6	3	a_m	Accelerometers	m/s^2
7:9	3	ω_{fix}	Rates when fixed to Earth	rad/s
10:12	3	v_{fix}	Velocity when fixed to Earth (ECF)	m/s
13:15	3	τ_m	Torque disturbance (0 in freefall)	rad/s
16:18	3	f_m	Force disturbance (0 in freefall)	m/s^2
19:21	3	p_{gps}	GPS position (ECF)	m
22:24	3	v_{gps}	GPS velocity (ECF)	m/s

Table 2-3: Filter Control Input

Index	Size	Symbol	Description	Units
1:3	3	τ	Specific torque	rad/s^2
4:6	3	f	Specific force	m/s^2
7	1	θ_e	Angle from ECI to ECF	rad

2.3.1 Linearization

We now proceed to derive the specific equations for this Kalman filter, using our choice of state variables and measurements. Nonlinear equations have been derived in the navigation theory of Section 2.1. These equations must now be linearized for an Extended Kalman Filter using the procedures described in Section 2.2.2. This is done be analyzing the dynamics of the estimation error, which is linear for small error. A linear relation between the measurement error and the state error is also established. The final results are summarized in Section 2.3.2.

2.3.1.1 Dynamics

Analysis begins with linearization of the dynamics. The full equations of navigation are converted into a linear state model.

Attitude Representation

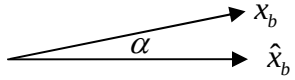

Figure 2-8: Small Angle Orientation Error

Special care is taken in the treatment of angles. In particular, we wish to represent the error in the four quaternion parameters using three small angles. This is possible because the quaternions are non-minimal.

Similar to the error states discussed earlier, let α be the small angle rotation from the estimated orientation $\hat{q}$ to the actual orientation q. The relation between a body vector x_b at estimated and actual orientation is shown in Figure 2-8.

Accordingly, the relation between the estimated and actual DCM is given by

$$B(q) = R_\alpha B(\hat{q}), \tag{2-292}$$

where R_α is the small angle rotation matrix defined by (2-74). In other words, transformation into the actual body coordinates can be performed using a rotation of $B(\hat{q})$ followed by a rotation of R_α.

The quaternions may be propagated using the state estimates:

$$\dot{\hat{q}} = f_q(\hat{q})\hat{\omega}. \tag{2-293}$$

Careful consideration will reveal the resulting dynamics of α. From the definition, the derivative of α is related to ω. In a manner of thinking, $\dot{\alpha} \approx \omega$. Since we are actually using α to represent angular error, the rate in this expression must be replaced by a rate difference:

$$\dot{\alpha} = \omega - \hat{\omega}' . \tag{2-294}$$

For comparison, both rates must be expressed in the same frame. The first rate, ω, is the actual body rate expressed in the actual body frame. The second rate, $\hat{\omega}'$, is the estimate body rate expressed in the actual body frame. This is related to the estimated body rate in the estimate body frame by the small rotation

$$\hat{\omega}' = R_\alpha \hat{\omega} . \tag{2-295}$$

Collecting the previous equations, we have

$$\dot{\alpha} = \omega - R_\alpha \hat{\omega} . \tag{2-296}$$

Reduction using the error state $\delta\omega$ yields

$$\dot{\alpha} = \hat{\omega} + \delta\omega - (I - \alpha\times)\hat{\omega} = \delta\omega + \alpha \times \hat{\omega} . \tag{2-297}$$

Reversing the cross product gives

$$\dot{\alpha} = -\hat{\omega} \times \alpha + \delta\omega , \tag{2-298}$$

which is linear and suitable for the Kalman filter. Note that we can determine α from $\hat{q}$ and q by calculating

$$R_\alpha = B(q)B(\hat{q})^T , \tag{2-299}$$

and then extracting α from the skew-symmetric part. The Kalman measurement update can be modified so that corrections in α are passed directly to the quaternions. Consider the quaternion propagation over a short period of time

$$\dot{\hat{q}}\delta t = f_q(\hat{q})\hat{\omega}\delta t . \tag{2-300}$$

Since $\hat{\omega}\delta t \approx \delta\alpha$, the quaternion correction is related to the small angle correction by

$$\delta\hat{q} = f_q(\hat{q})\delta\alpha . \tag{2-301}$$

This can be written in the form of the Kalman measurement update:

$$\hat{q}_+ = \hat{q}_- + f_q(\hat{q}_-)i_\alpha , \tag{2-302}$$

where i_α is the α portion of the innovations vector. In other words, this portion of the innovations must be separated and processed differently from the other state variables. This allows the three dimensional small angle misalignment to be offloaded to the four dimensional quaternions.

Rotational Dynamics

We now linearize the rotational dynamics

$$\dot{\omega} = -J^{-1}\left(\omega \times \left(J\omega\right)\right) + \tau_d + \tau\,, \tag{2-303}$$

where the specific torque input has been divided into a known part τ and a disturbance τ_d. We wish to find the propagation of the rate error $\delta\omega$. Using the estimate propagation

$$\dot{\hat{\omega}} = -J^{-1}\left(\hat{\omega} \times \left(J\hat{\omega}\right)\right) + \hat{\tau}_d + \tau\,, \tag{2-304}$$

the error dynamics are

$$\delta\dot{\omega} = \dot{\omega} - \dot{\hat{\omega}} = -J^{-1}\left((\hat{\omega} + \delta\omega) \times \left(J(\hat{\omega} + \delta\omega)\right) - \hat{\omega} \times (J\hat{\omega})\right) + \delta\tau_d\,. \tag{2-305}$$

The cross product can be distributed and the small term $\delta\omega \times J\delta\omega$ can be ignored, producing

$$\delta\dot{\omega} \approx J^{-1}\left([\hat{\omega}\times]J - [(J\hat{\omega})\times]\right)\delta\omega + \delta\tau_d\,. \tag{2-306}$$

Translational Dynamics

The position equation $\dot{p} = v$ is already linear, so $\delta\dot{p} = \delta v$. The velocity equation

$$\dot{v} = B^T(q)\left(f + f_d\right) + g(p) - b_g \tag{2-307}$$

requires more attention. We wish to find the propagation of the velocity error δv in terms of the other state estimation errors. Using the estimate propagation

$$\dot{\hat{v}} = B^T(\hat{q})\left(f + \hat{f}_d\right) + g(\hat{p}) - \hat{b}_g\,, \tag{2-308}$$

the error dynamics are

$$\delta\dot{v} = B^T(\hat{q})(I + [\alpha\times])\left(f + \hat{f}_d + \delta f_d\right) + g(p) - b_g - B^T(\hat{q})\left(f + \hat{f}_d\right) - g(\hat{p}) + \hat{b}_g\,, \tag{2-309}$$

which reduces to

$$\delta\dot{v} = B^T(\hat{q})[\alpha\times](f + \hat{f}_d + \delta f_d) + B^T(\hat{q})\delta f_d + g(p) - g(\hat{p}) - \delta b_g\,. \tag{2-310}$$

The difference in gravity can be approximated using the gravity gradient:

$$g(p) - g(\hat{p}) \approx \left.\frac{\partial g}{\partial p}\right|_{\hat{p}} \delta p\,, \tag{2-311}$$

which will be noted G_p. The cross product can be reversed and small terms dropped, yielding

$$\delta\dot{v} \approx -B^T(\hat{q})[(f + \hat{f}_d)\times]\alpha + B^T(\hat{q})\delta f_d + G_p\delta p - \delta b_g\,. \tag{2-312}$$

2.3.1.2 Measurements

We now proceed to model and linearize the measurements. Each measurement will be pre-compensated using the state estimate, as described in Section 2.2.2.2. This leads to an expression that relates the state estimation error to the measurement error. For small error, this relation is linear and provides the measurement matrix *H* for the Kalman filter.

Gyroscopes

Gyroscopes that are mounted to the vehicle provide a direct measurement of body rate. A simple sensor model includes a bias and noise term:

$$\omega_m = \omega + b_\omega + n_\omega\,. \tag{2-313}$$

This measurement can be corrected using the state estimate, leading to the small measurement

$$\delta\omega_m = \omega_m - \hat{b}_\omega - \hat{\omega} = \delta\omega + \delta b_\omega + n_\omega\,. \tag{2-314}$$

Accelerometers

The accelerometer measurement can be written as

$$a_m = B(q)\left(\dot{v} - g(p) + b_g\right) + b_a + n_a\,, \tag{2-315}$$

which is the difference between inertial and gravitational acceleration in body coordinates, with biases and additive noise. We can eliminate $\dot{v}$ from this equation by substituting the process equation

$$\dot{v} = B^T(q)\left(f + f_d\right) + g(p) - b_g\,. \tag{2-316}$$

The result simplifies to

$$a_m = f + f_d + b_a + n_a\,. \tag{2-317}$$

Compensating gives the small measurement

$$\delta a_m = a_m - \hat{b}_a - f - \hat{f}_d \approx \delta f_d + \delta b_a + n_a \tag{2-318}$$

Fixed-to-Earth

When the vehicle is fixed to Earth, the motion matches that of the Earth's surface. We can define a measurement that is the difference between the actual body rate and the body rate that the Earth would impose:

$$\omega_{fix} = \omega - B(q)\omega_e + n_{\omega fix} = 0\,. \tag{2-319}$$

A noise term is included to model vehicle vibrations. This can be expanded using the error states

$$\omega_{fix} = \omega - R_\alpha B(\hat{q})\omega_e + n_{\omega fix} = \omega - \left(I - [\alpha\times]\right)B(\hat{q})\omega_e + n_{\omega fix}\,. \tag{2-320}$$

Substituting the estimated measurement and reversing the cross product produces

$$\delta\omega_{fix} = \omega_{fix} - \hat{\omega} + B(\hat{q})\omega_e = -[B(\hat{q})\omega_e\times]\alpha + \delta\omega + n_{\omega fix}\,. \tag{2-321}$$

When fixed to Earth, the ECF velocity is zero, that is

$$v_{fix} = -\omega_e \times R_e p + R_e v + n_{vfix} = 0 \tag{2-322}$$

Again, a noise term is included to model vibrations. Expression in terms of the error states leads to

$$\delta v_{fix} = v_{fix} + \omega_e \times R_e \hat{p} - R_e \hat{v} = -\omega_e \times R_e \delta p + R_e \delta v + n_{fix} \tag{2-323}$$

Disturbance (Freefall)

If the disturbances are somehow known or measured, we have

$$\tau_m = \tau_d \tag{2-324}$$

$$f_m = f_d \tag{2-325}$$

This is linear, and the small measurements are simply

$$\delta\tau_m = \tau_m - \hat{\tau}_d = \delta\tau_d \tag{2-326}$$

$$\delta f_m = f_m - \hat{f}_d = \delta f_d \tag{2-327}$$

In freefall, we have $\tau_m = f_m = 0$, assuming that the control input is zero.

GPS

A GPS receiver typically outputs position and velocity in the ECF frame. Similar to v_{fix}, these ECF quantities can be expressed in terms of the ECI states:

$$p_{gps} = R_e p + n_{pgps} \tag{2-328}$$

$$v_{gps} = R_e v - \omega_e \times R_e p + n_{vgps}. \tag{2-329}$$

These measurements can then be compensated to produce small measurement errors:

$$\delta p_{gps} = p_{gps} - R_e \hat{p} = R_e \delta p + n_{pgps} \tag{2-330}$$

$$\delta v_{gps} = v_{gps} + \omega_e \times R_e \hat{p} - R_e \hat{v} = -\omega_e \times R_e \delta p + R_e \delta v + n_{vgps} \tag{2-331}$$

2.3.1.3 Noise Modeling

Since the Kalman filter requires white noise, the time-correlation in disturbances and biases must be modeled using additional state variables.

Torque and Force Disturbances

The torque and force disturbances are modeled using a first order linear filter with white input:

$$\dot{\tau}_d = -r_\tau \tau_d + w_\tau \tag{2-332}$$

$$\dot{f}_d = -r_f f_d + w_f. \tag{2-333}$$

This produces an exponential autocorrelation function as described in Section 2.2.5.2, with a steady state autocorrelation function of

$$E\{c(t)c(t-T)\} = \frac{\sigma_w^2}{2r} e^{-2rT}. \tag{2-334}$$

The variance is given by evaluating (2-334) at $T = 0$, which produces

$$\sigma^2 = \frac{\sigma_w^2}{2r} \tag{2-335}$$

Additional white noise in the torque disturbance can be included using the body rate process noise w_ω. This does not work for the force disturbance because any force noise is measured directly by the IMU.

Bias Dynamics

The biases are similar to the disturbance dynamics. The white component of their noise appears as the measurement noise. The bias states represent the colored part, and have the following first order dynamics:

$$\dot{b}_\omega = -r_\omega b_\omega + w_{b\omega} \tag{2-336}$$

$$\dot{b}_a = -r_a b_a + w_{bg} \tag{2-337}$$

$$\dot{b}_g = -r_g b_g + w_{bg} . \tag{2-338}$$

The steady state autocorrelation function is once again

$$E\{c(t)c(t-T)\} = \frac{\sigma_w^2}{2r} e^{-2rT} . \tag{2-339}$$

However, the gravity model accuracy degrades with distance instead of time. Since

$$\frac{db_g}{dt} = \frac{\partial b_g}{\partial p_{ecf}} \frac{dp_{ecf}}{dt} = \frac{\partial b_g}{\partial p_{ecf}} v_{ecf} , \tag{2-340}$$

The variance for the noise to the gravity bias can be calculated using

$$E\{w_{bg} w_{bg}^T\} = E\left\{ \frac{\partial b_g}{\partial p_{ecf}} v_{ecf} v_{ecf}^T \left(\frac{\partial b_g}{\partial p_{ecf}} \right)^T \right\} . \tag{2-341}$$

This can be simplified by assuming symmetry in the gravity model error or by using only velocity magnitude. Alternatively, it can be expanded and calculated in full using the state estimate.

Note that this technique does not attempt to remember the gravity model bias as a function of position. For circular flight paths, the vehicle may return to a position where the bias is well known. Unfortunately, the EKF does not provide a mechanism to incorporate this. Instead, motion is any direction is taken to increase the uncertainty in the gravity model bias estimate.

2.3.2 Summary of Equations

From the previous selection of states and linearization, the equations for the Kalman filter and for the full nonlinear state estimator can be summarized as follows.

Nonlinear Dynamics

The fundamental navigation states are modeled using the following nonlinear differential equations:

$$\dot{q} = f_q(q)\omega \tag{2-342}$$

$$\dot{\omega} = -I^{-1}\left(\omega \times (I\omega)\right) + \tau_d + \tau \tag{2-343}$$

$$\dot{p} = v \tag{2-344}$$

$$\dot{v} = B(q)^T\left(f + f_d\right) + g(p) - b_g \tag{2-345}$$

All additional states, including disturbances and biases, are modeled as simple decays:

$$\dot{x}_i = -r_i x_i \,. \tag{2-346}$$

Linearized Small Error Dynamics

The nonlinear propagation produces small state error dynamics that can be approximated using the linear system

$$\delta\dot{x} \approx A(\hat{x}, u, t)\delta x \,. \tag{2-347}$$

The linearized equations for the core navigation states are

$$\dot{\alpha} = \left[-\hat{\omega}\times\right]\alpha + \delta\omega \tag{2-348}$$

$$\delta\dot{\omega} = J^{-1}\left([\hat{\omega}\times]J - [(J\hat{\omega})\times]\right)\delta\omega + \delta\tau_d \tag{2-349}$$

$$\delta\dot{p} = \delta v \tag{2-350}$$

$$\delta\dot{v} = B(\hat{q})^T[(f + \hat{f}_d)\times]\alpha + B(\hat{q})^T\delta f_d + G_p\delta p - \partial b_g \,, \tag{2-351}$$

and the remaining disturbances and biases follow

$$\delta\dot{x}_i = -r_i\delta x_i \tag{2-352}$$

These equations can be assembled into the following *A* matrix:

$$\begin{pmatrix} \dot{\alpha} \\ \delta\dot{\omega} \\ \delta\dot{\tau}_d \\ \delta\dot{p} \\ \delta\dot{v} \\ \delta\dot{f}_d \\ \delta\dot{b}_\omega \\ \delta\dot{b}_a \\ \delta\dot{b}_g \end{pmatrix} \approx \begin{bmatrix} [-\hat{\omega}\times] & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_{\omega\omega} & I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -r_{\tau d} I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ A_{v\alpha} & 0 & 0 & G_p & 0 & \hat{B}^T & 0 & 0 & -I \\ 0 & 0 & 0 & 0 & 0 & -r_{fd} I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -r_{b\omega} I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -r_{ba} I & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -r_{bg} I \end{bmatrix} \begin{pmatrix} \alpha \\ \delta\omega \\ \delta\tau_d \\ \delta p \\ \delta v \\ \delta f_d \\ \delta b_\omega \\ \delta b_a \\ \delta b_g \end{pmatrix} \quad (2\text{-}353)$$

All terms in this matrix are 3 by 3, including the identity matrix *I* and zero matrix 0. The additional terms are

$$A_{\omega\omega} = J^{-1}\left([\hat{\omega}\times]J - [(J\hat{\omega})\times]\right) \quad (2\text{-}354)$$

$$A_{v\alpha} = -\hat{B}^T\left[\left(f + \hat{f}_d\right)\times\right]. \quad (2\text{-}355)$$

Measurement Compensation and Linearization

Measurements are compensated as follows, producing small error measurements:

$$\delta\omega_m = \omega_m - \hat{b}_\omega - \hat{\omega} \quad (2\text{-}356)$$

$$\delta a_m = a_m - \hat{b}_a - f - \hat{f}_d \quad (2\text{-}357)$$

$$\delta\omega_{fix} = B(\hat{q})\omega_e - \hat{\omega} \quad (2\text{-}358)$$

$$\delta v_{fix} = \omega_e \times R_e\hat{p} - R_e\hat{v} \quad (2\text{-}359)$$

$$\delta\tau_m = \tau_m - \tau - \hat{\tau}_d \quad (2\text{-}360)$$

$$\delta f_m = f_m - f - \hat{f}_d \quad (2\text{-}361)$$

$$\delta p_{gps} = p_{gps} - R_e\hat{p} \quad (2\text{-}362)$$

$$\delta v_{gps} = v_{gps} + \omega_e \times R_e\hat{p} - R_e\hat{v}\,. \quad (2\text{-}363)$$

The small measurements can then be approximated by the linear form $\delta z = H\delta x$, to be used in the Kalman filter. The resulting matrix expression for H is

$$\begin{pmatrix} \delta\omega_m \\ \delta a_m \\ \delta\omega_{fix} \\ \delta v_{fix} \\ \delta\tau_m \\ \delta f_m \\ \delta p_{gps} \\ \delta v_{gps} \end{pmatrix} = \begin{bmatrix} 0 & I & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 & I & 0 \\ \left[-\hat{B}\omega_e \times\right] & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\omega_e \times R_e & R_e & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & R_e & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\omega_e \times R_e & R_e & 0 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} \alpha \\ \delta\omega \\ \delta\tau_d \\ \delta p \\ \delta v \\ \delta f_d \\ \delta b_\omega \\ \delta b_a \\ \delta b_g \end{pmatrix}. \quad \text{(2-364)}$$

2.3.3 Navigation Observability

The linearized dynamics, which are used by the Kalman filter, form an LTV system that can be analyzed for observability. While at this point we have already selected the state variables, this can provide insight into filter convergence and highlight areas of weak measurement. These results will also aid in the interpretation of covariance analysis that is conducted in Section 3.3 of this report.

Full LTV observability analysis through computation of the state transition matrix is difficult. Instead, we begin with the LTI observability test matrix, which is straightforward to calculate but will usually underestimate observability. States that are unobservable by this method will then be viewed from an LTV sense. The LTV analysis is not completely rigorous, but the results are verified in covariance analysis and full simulation.

This method requires the assumption that the LTV system is LTI over short periods of time, as was discussed in Section 2.2.4.1. This assumption is met if the LTI approximation is valid for a time period spanning n measurements, where n is the number of state variables. This requirement can be reduced when the sampling frequency is fast compared to the system dynamics. Any restrictions in this assumption could probably be overcome by performing the complete LTV observability analysis.

As defined in Section 2.2.4.1, the LTI observability test matrix is given by

$$M_{LTI} = \begin{pmatrix} H \\ HA \\ \vdots \\ HA^{n-1} \end{pmatrix}. \quad \text{(2-365)}$$

We will start with only the core measurements: gyroscopes $\delta\omega_m$, accelerometers δa_m, and GPS position δp_{gps}. These are the linearized small measurement errors that are given to the Kalman filter. The effect of additional measurements will be added later, and provides some additional observability.

The LTI observability test matrix calculation can be conducted using the 3 by 3 blocks that were used to express the *A* and *H* matrices. The first 12 block rows have been calculated below:

$$\begin{bmatrix} H \\ HA \\ HA^2 \\ HA^3 \end{bmatrix} = \left[\begin{array}{ccccccccc} 0 & I & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 & I & 0 \\ 0 & 0 & 0 & R_e & 0 & 0 & 0 & 0 & 0 \\ \hdashline 0 & A_{\omega\omega} & I & 0 & 0 & 0 & -r_{b\omega} I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -r_{fd} I & 0 & -r_{ba} I & 0 \\ 0 & 0 & 0 & 0 & R_e & 0 & 0 & 0 & 0 \\ \hdashline 0 & A_{\omega\omega}^2 & M_1 & 0 & 0 & 0 & r_{b\omega}^2 I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_{fd}^2 I & 0 & r_{ba}^2 I & 0 \\ R_e A_{v\alpha} & 0 & 0 & R_e G_p & 0 & R_e B^T(\hat{q}) & 0 & 0 & -R_e \\ \hdashline 0 & A_{\omega\omega}^3 & M_2 & 0 & 0 & 0 & r_{b\omega}^3 I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_{fd}^3 I & 0 & r_{ba}^3 I & 0 \\ -\hat{\omega} \times R_e A_{v\alpha} & R_e A_{v\alpha} & 0 & 0 & R_e G_p & -r_{fd} R_e B^T(\hat{q}) & 0 & 0 & r_{bg} R_e \end{array}\right] \tag{2-366}$$

where

$$M_1 = A_{\omega\omega} - r_{\tau d} I \tag{2-367}$$

$$M_2 = A_{\omega\omega}^2 - r_{\tau d} A_{\omega\omega} + r_{\tau d}^2 I \,. \tag{2-368}$$

Each block column corresponds to a 3 dimensional state variable vector, such as α or ω. The observability of an individual state can be accessed in two steps. First, the block column must be linearly independent from the other block columns. Second, the block column itself must have three independent columns. The number of independent columns in a matrix is equal to its rank. The matrices *I*, R_e, and B^T are invertible, so they have 3 independent columns.

We will start by finding the observability without the bias states, which is done by ignoring the rightmost three block columns. Analysis can proceed row by row in blocks, with care not to reuse rows in the argument for linear independence.

The first three rows correspond to the raw measurements. Row 1 (ω_m) makes column 2 independent (ignoring bias columns), so ω is observable through the gyroscopes. Row 2 (a_m)

makes column 6 independent, so f_d is observable through the accelerometers. Row 3 (p_{gps}) makes column 4 independent, so p is observable through the GPS position. Note that R_e and I are invertible, so these matrices in turn have independent columns, and therefore each element of ω, f_d, and p is observable.

The next three rows correspond to the measurements passed once through the system. In row 4, t_d (column 3) becomes observable as derivative of body rates. Row 5 yields nothing here but will be revisited later. Row 6 observes v (column 5) as the derivate of position.

Continuing, the next three rows correspond to a second propagation of measurements through the system. Row 7 and 8 yield nothing here. Row 9 produces a term for attitude error α (column 1):

$$R_e A_{v\alpha} = -R_e B(\hat{q})^T \left[\left(f + \hat{f}_d\right)\times\right] \tag{2-369}$$

As rotation matrices, R_e and $B(\hat{q})^T$ are invertible. However, the remaining cross-product matrix is singular, since (as with any vector)

$$\left(f + \hat{f}_d\right)\times\left(f + \hat{f}_d\right) = 0\,. \tag{2-370}$$

In other words, attitude cannot be resolved about the direction of external force. This would correspond to compass heading for a vehicle at rest on the Earth, because the vehicle is being supported vertically by the ground. If the vehicle is in freefall, the force is zero and this term has rank 0, so no attitude can be observed. Otherwise, there are typically two dimensions of attitude observability. The additional term in row 12 contains the same null space.

The observability of attitude is increased by time variation of the force $f + \hat{f}_d$. We can proceed to show what type of force variation is needed for observability of the LTV system. The dynamics of α are simplified when viewed in inertial coordinates:

$$\frac{d}{dt}\left(B^T\alpha\right) = \dot{B}^T\alpha + B^T\dot{\alpha} = B^T\left[\omega\times\right]\alpha + B^T\left(-\left[\hat{\omega}\times\right]\alpha + \delta\omega\right) \approx B^T\delta\omega\,. \tag{2-371}$$

The small term $\delta\omega\times\alpha$ was ignored. This leads to the solution

$$B^T\alpha = B_o^T\alpha_o + \int_{t_o}^{t} B^T\delta\omega dt\,, \tag{2-372}$$

and

$$\alpha = BB_o^T\alpha_o + B\int_{t_o}^{t} B^T\delta\omega dt\,. \tag{2-373}$$

As a result, the α portion of the state transition matrix is given by

$$\Phi_{\alpha\alpha}(t,t_o) = BB_o^T . \tag{2-374}$$

An inertially constant force f_I (in ECI) will produce a body force given by

$$f + \hat{f}_d = Bf_I . \tag{2-375}$$

Recall from the LTV observability analysis that the LTV observability matrix will contain rows given by $M_{LTI}(t)\Phi(t,t_o)$. With the inertial force definition, the attitude term of (2-369) becomes

$$R_e A_{v\alpha} \Phi_{\alpha\alpha}(t,t_o) = -R_e \hat{B}^T \left[(Bf_I)\times\right] BB_o^T \tag{2-376}$$

in the LTV observability matrix. The null space of this term is $B_o f_I$, since

$$\left(\left[(Bf_I)\times\right] BB_o^T\right) B_o f_I = \left[(Bf_I)\times\right] Bf_I = 0 . \tag{2-377}$$

This null space changes only due to f_I, and is unaffected by B. As a result, pure body rotation does not produce attitude observability. Variation in the inertial direction of force is required.

The observability of biases is a second matter. These states are stable, because of their proportional decay, so the system is detectable regardless of their observability. Still, they can be observed in many cases.

We must revisit row 1, because the gyro bias (column 7) makes the body rates (column 2) no longer unique. These two states can be discriminated using row 12, with the same limitations as the observation of attitude. Note that the biases dynamics are a simple exponential decay. As a result, the LTV observability analysis shows that any change in $A_{v\alpha}$ will produce full bias observability. This includes rotating the vehicle.

There may also be information in row 7 with the term

$$A_{\omega\omega} = J^{-1}\left([\hat{\omega}\times]J - [(J\hat{\omega})\times]\right). \tag{2-378}$$

This term is rank 0 under 3-axis mass symmetry ($J = I$) or when $\hat{\omega} = 0$. It is rank 2 when rotating about a principal axis of inertia. Otherwise, the precession of the rotating body can be used to fully differentiate between real rotation and biased measurement.

For analysis of biases, we must also revisit row 2 because the force disturbance (column 6) and accelerometer bias (column 8) are not independent in this row. They can be discriminated by their decay rates in row 5 or row 8, but this may be unreliable if both exhibit slow dynamics. There is one dimension available in row 9, because only two

dimensions of attitude were observed. Row 9 is also the only means of observing the gravity bias (column 9).

The dynamics of theses biases are also simple decays, which simplify LTV observability analysis. The accelerometer bias b_a rotates with the vehicle while the gravity model bias b_g is inertially fixed. As a result, b_a can be observed when there is sufficient variation in the vehicle orientation $B^T(\hat{q})$, while b_g requires variation in the inertial direction of force.

The ambiguity between attitude errors and biases is depicted in Figure 2-9. This effect is present even if bias states are omitted from the model. Attitude accuracy is improved by proper bias compensation.

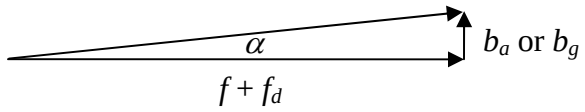

Figure 2-9: Equivalence of a Bias and a Small Angle Error

Rows beyond what was calculated in (2-366) do not provide any significant additional information. The left three columns are isolated by the nature of the A matrix. The other columns contain small terms, such as the gravity gradient G_p and the decay rates.

We can consider the effect of additional measurements. An ECF velocity measurement, such as v_{fix} or v_{gps}, does not increase in observability. The small Earth rate term $\omega_e\times$ will appear in various columns, including the attitude column, but this is redundant to the velocity information that is already provided from the position measurement. Still, these measurements can increase the filter response and accuracy when available.

The measurement ω_{fix} greatly improves observability. The first two rows that this measurement adds to the observability test matrix are

$$\begin{bmatrix} H_{fix} \\ H_{fix}A \end{bmatrix} = \begin{bmatrix} \left[-\hat{B}\omega_e\times\right] & I & 0 & 0 & 0 & 0 & 0 & 0 \\ -\hat{B}\left[\omega_e\times\right]\left[\hat{\omega}\times\right] & \left[-\hat{B}\omega_e\times\right]+A_{\omega\omega} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \qquad (2\text{-}379)$$

The first row here gives provides bias free measurement of body rates, which leads to continuous observability of the gyroscope biases. In addition, the $\hat{B}\omega_e\times$ terms in both rows produce more observability of attitude by aligning the Earth rate vector. These terms are of rank 2, and are very small, but may be visible with high quality gyroscopes.

Higher order compensation of the IMU, such as scaling error states, will not be observable in the LTI sense because their effect is the same as that of biases. However, under rich motion where the force and body rates vary greatly, it may be possible to observe these states in the LTV system.

The measurement of force disturbance provides full observability of the accelerometer bias. The measurement of torque disturbance does not add observability, but can increase performance when the measurement of the disturbance is of higher quality than the gyroscopes. These disturbance measurements can reduce IMU drift during freefall, when observability is poor.

In summary, this observability analysis has reached several important conclusions. Body rates, position, velocity, and force and torque disturbances are always observable. Attitude and gravity modeling errors are observable if the inertial direction of force is varied sufficiently. IMU biases are observable if the IMU is rotated. Additional gyroscope observability comes from the fixed-to-Earth measurements and from the dynamics of rotation.

2.3.4 Additional Considerations

Some additional techniques are important for the implementation of the Kalman Filter in navigation. The topics discussed below are not pursued fully in this project, but present interesting and relevant material.

Offset GPS Antenna

The GPS antenna is usually located on the extremity of the vehicle and can therefore be a large distance away from the center of mass. This can be incorporated into the filter measurement equations so that the solution is exact.

Consider an antenna at offset *l*, which is a constant vector in body coordinates. The position and velocity of the antenna in ECI coordinates are given by

$$p_l = p + B^T l \tag{2-380}$$

$$v_l = v + B^T(\omega \times l). \tag{2-381}$$

The GPS solution is a measure of antenna position and velocity in ECF coordinates:

$$p_{gps} = R_e p_l = R_e\left(p + B^T l\right) \tag{2-382}$$

$$v_{gps} = R_e v_l - \omega_e \times R_e p_l = R_e\left(v + B^T(\omega \times l)\right) - \omega_e \times R_e\left(p + B^T l\right). \tag{2-383}$$

The noise terms have been ignored in this analysis, but are present throughout. This can be rearranged as

$$p_{gps} = R_e p + R_e B^T l \tag{2-384}$$

$$v_{gps} = R_e v - \omega_e \times R_e p + R_e B^T(\omega \times l) - \omega_e \times R_e B^T l\,. \tag{2-385}$$

We will now express these measurements in terms of the error states. Since

$$B^T = \hat{B}^T R_\alpha^T = \hat{B}^T \left(I + \alpha\times\right) \tag{2-386}$$

we can write the following:

$$B^T l = \hat{B}^T \left(I + \alpha\times\right) l = \hat{B}^T l - \hat{B}^T \left(l\times\alpha\right) \tag{2-387}$$

and

$$B^T \left(\omega\times l\right) = \hat{B}^T \left(I + \alpha\times\right)\left(\left(\hat{\omega} + \delta\omega\right)\times l\right) = \hat{B}^T \left(\left(\hat{\omega} + \delta\omega\right)\times l\right) - \hat{B}^T \left(\left(\hat{\omega} + \delta\omega\right)\times l\right)\times\alpha \tag{2-388}$$

The small measurements are then

$$\delta p_{gps} = p_{gps} - R_e \hat{p} - R_e \hat{B}^T l = -R_e \hat{B}^T \left[l\times\right]\alpha + R_e \delta p \tag{2-389}$$

and

$$\begin{aligned} \delta v_{gps} &= v_{gps} - R_e \left(v + \hat{B}^T \left(\hat{\omega}\times l\right)\right) + \omega_e \times R_e \left(p + \hat{B}^T l\right) \\ &= \left(\left[\omega_e \times\right] R_e \hat{B}^T \left[l\times\right] - R_e \hat{B}^T \left[\left(\hat{\omega}\times l\right)\times\right]\right)\alpha - R_e \hat{B}^T \left[l\times\right]\delta\omega - \left[\omega_e \times\right] R_e \delta p + R_e \delta v . \end{aligned} \tag{2-390}$$

This gives the new formula for measurement compensation and the *H* matrix in the filter. The observability of the system may have improved because of the new α term $R_e \hat{B}^T \left[l\times\right]\alpha$ in δp_{gps}. This arises because an attitude error will now cause a position error. Looking back at the observability test matrix (2-366), these two errors are indistinguishable when attitude is constant. However, changes in attitude will aid attitude alignment, because of the change in antenna position. This is only effective when the antenna offset is large compared to the GPS accuracy.

Note also that changes in *l* will also provide attitude observability. This would be the case if the GPS receiver switched between multiple antennas at different locations. Multiple GPS receivers would also produce this effect. This would only be effective on structures that are large compared to the GPS accuracy.

The GPS velocity equations also contain α. In theory, this information is redundant to the GPS position information. The term $R_e \hat{B}^T \left[\left(\hat{\omega}\times l\right)\times\right]\alpha$ shows that GPS velocity will also be affected by an attitude error.

Offset Accelerometers

There may also be an offset between the IMU and the center of mass. This has no effect on the gyroscopes, but will add centripetal acceleration to accelerometer measurement. For an accelerometer mounted at a constant offset *l*, the measured acceleration is

$$a_m = f + f_d + b_a + \dot{\omega} \times l + \omega \times (\omega \times l). \tag{2-391}$$

This can be expanded in terms of the error states. We can begin with

$$\dot{\omega} = \tau + \tau_d - J^{-1}(\omega \times (I\omega)) = \tau + \hat{\tau}_d + \delta\tau_d - J^{-1}((\hat{\omega} + \delta\omega) \times (J(\hat{\omega} + \delta\omega))). \tag{2-392}$$

For small errors, this reduces to

$$\dot{\omega} \approx \tau + \hat{\tau}_d + \delta\tau_d - J^{-1}(\hat{\omega} \times (J\hat{\omega})) - J^{-1}([\hat{\omega} \times] J - [J\hat{\omega} \times])\delta\omega, \tag{2-393}$$

which produces

$$\dot{\omega} \times l = -l \times \dot{\omega} \approx -l \times (\tau + \hat{\tau}_d) - l \times \delta\tau_d + l \times J^{-1}(\hat{\omega} \times (J\hat{\omega})) - l \times J^{-1}([\hat{\omega} \times] J - [J\hat{\omega} \times])\delta\omega. \tag{2-394}$$

The centripetal term can be expanded as well:

$$\omega \times (\omega \times l) = (\hat{\omega} + \delta\omega) \times ((\hat{\omega} + \delta\omega) \times l) \approx \hat{\omega} \times (\hat{\omega} \times l) + \hat{\omega} \times (\delta\omega \times l) + \delta\omega \times (\hat{\omega} \times l). \tag{2-395}$$

The cross products can be reversed and grouped to produce the following:

$$\omega \times (\omega \times l) \approx \hat{\omega} \times (\hat{\omega} \times l) - ([\hat{\omega} \times][l \times] + [(\hat{\omega} \times l) \times])\delta\omega. \tag{2-396}$$

These results suggest the measurement compensation

$$\delta a_m = a_m - f - f_d - b_a + l \times (\tau + \hat{\tau}_d) - l \times J^{-1}(\hat{\omega} \times (J\hat{\omega})) - \hat{\omega} \times (\hat{\omega} \times l) \tag{2-397}$$

which produces the small measurement

$$\delta a_m = (l \times J^{-1}([\hat{\omega} \times] J - [J\hat{\omega} \times]) - [\hat{\omega} \times][l \times] - [(\hat{\omega} \times l) \times])\delta\omega - l \times \delta\tau_d + \delta f_d + \delta b_a. \tag{2-398}$$

This will increase the observability of body rates and gyroscope biases in a rotating vehicle.

This completes the theoretical discussion. We now proceed to investigate the filter performance in simulation.

3. Simulation

Numerical simulation has been conducted on a PC using software models of the filter and dynamics. This is a valuable tool for characterizing filter performance. It provides a large variety of test scenarios and methods of analysis. The filter robustness can be quantified and areas of weakness can be exposed. Simulation is the most comprehensive test of filter function.

The benefits of simulation can be categorized into several areas. First, it provides a way to verify that filter equations are derived and programmed correctly. Second, it confirms and extends the theoretical analysis. For example, the observability predictions are complemented by the simulation of the covariance. Finally, simulation provides a method of testing the filter performance under different operating conditions. Simulations have been conducted using both real and flight data, which provides both realistic and extreme testing environments.

This simulation discussion is divided into several areas. Section 3.1 gives an overview of the Simulation setup and process, describing the tools and datasets that are used. Section 3.2 describes the Simulink model that is used for simulation. This provides a summary of the software implementation of the filter algorithm.

Section 3.3 presents the results from covariance analysis, in which the filter covariance is propagated numerically using different trajectories, providing a theoretical estimate of filter accuracy. This tool extends the observability analysis by quantifying the effect of noise parameters and flight path richness. It also predicts the filter sensitivity to the noise of each sensor and to other parameters. This produces useful guidelines for system design and sensor selection.

Section 3.4 presents the result from full simulation, in which the filter operates on simulated data. This shows that actual filter convergence matches the covariance analysis. Section 3.5 incorporates real flight data into the simulation. In these studies, the filter is tested in a realistic operating environment. Finally, section 3.6 focuses on the performance of the nonlinear initializer, which uses gyrocompassing to bring the state estimate within the capture range of the Extended Kalman Filter.

3.1 Overview

This simulation provides a software implementation of the filter and the ability to test its performance. This requires routines to input the measurements and truth data. It also requires routines to analyze and plot the results.

Simulation is conducted in the MATLAB/Simulink environment, which are products of The MathWorks. This software provides a full set of mathematical and data analysis tools, including several differential equation solvers. It utilizes a graphical block diagram representation, which is very flexible and easy to document. This software is used both to

implement the filter itself and for auxiliary data processing. Simulation parameters, such as the noise covariances, are loaded from a script.

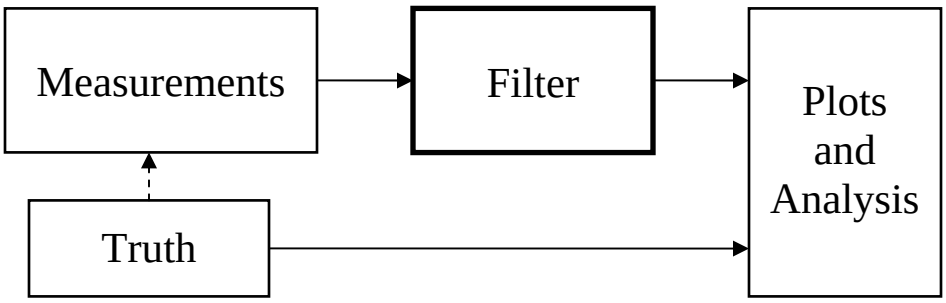

Figure 3-1: Simulation Block Diagram

The simulation itself consists of several components, as shown in Figure 3-1. Data flow begins with the measurements block, which contains the data from the IMU and GPS receiver as well as status lines indicating the arrival of new measurements. This data can be generated from four different sources:

1. Simulated measurements for an Earth stationary vehicle
2. Simulated measurements for a rigid body in motion
3. Recorded flight data from a small aircraft
4. Recorded flight data from a booster rocket

The flight data is available from current projects at MIT Lincoln Laboratory. The aircraft dataset contains measurements from a small airplane during takeoff, maneuvering, and landing. The booster rocket dataset contains measurement from a short range launch vehicle built by Orbital Systems Corporation.

The measurement data is processed by the filter, which is the Extended Kalman Filter described in detail in the Section 2.3. This block contains subsystems for propagating the full nonlinear state, linearizing about the estimated state, and applying the Kalman filter update. It also includes provisions for initialization and for calculating additional navigation outputs, such as geodetic coordinates. This block is described in more detail in Section 3.2.

The truth data is used as a reference to assess filter performance. The truth data is generated differently depending on the measurement set. For simulated measurements (sets 1 and 2 above), the truth data is arbitrary and is actually used to generate the measurements. The truth data for an Earth stationary vehicle is calculated using the Earth's rate of rotation. For simulated rigid body motion, the force and torque are input to a dynamic model which calculates the truth data. For real flight data, the truth data is taken from another navigation system, either onboard or post-processed.

In comparison, the simulated measurements have the advantage that the truth data is precisely known and controlled. It is possible to exhaustively test the effect of all parameters. However, this simulated data uses many of the same models as the filter, so it is

possible for modeling errors to go undetected. The real flight data contains the actual noise and dynamics, so it will uncover modeling errors. However, many parameters cannot be manipulated when using these datasets.

Output from the filter and truth model are analyzed and plotted in the final block. Most data is also logged to a MATLAB workspace where it can be archived and processed later using scripts. This allows multiple simulations to be compared.

The simulation produces several results of interest, but the attitude and position accuracy are of primary importance. This accuracy can be assessed using several different methods. The most direct measure is to compare the filter output to the truth data. However this assumes that the truth data is, in fact, correct. For the recorded flight data, the "truth" data does contain glitches and accuracy limitations.

A second reference is the pure inertial solution which can be calculated directly from the IMU. This solution is numerically robust and accurately represents the high frequency dynamics. However, this solution is prone to drift over long periods of time. The pure inertial solution is therefore best used as a short-term reference. The filter estimate should mimic the inertial solution, but may be offset.

Additional evaluation comes from the Kalman filter itself. The covariance gives indication of the degree of observability of the system and should converge to acceptable values. The variance can also be compared with the actual estimation error. The chi-squared statistic provides a measure of filter health. In effect, it gives the improbability of each measurement. Its value should remain reasonable, and will spike when a measurement does not fit the model.

3.2 Filter Implementation

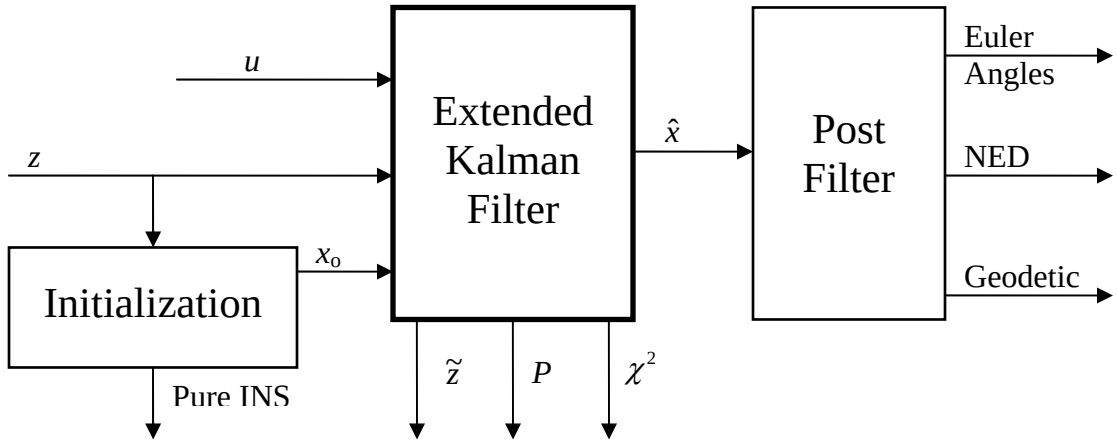

Figure 3-2: Top Level Filter Architecture

The specific filter implementation is summarized here. This describes the flow of information between filter components and outlines where specific calculations take place. It is representative of the actual Simulink model.

The top level filter architecture is shown is Figure 3-2 and consists of three main components. The Initialization block performs startup initialization and is responsible for calculating the initial state vector x_o. This block can provide an instant initialization, where initial values are simply converted to the coordinate system of the state vector. Alternatively, this block can use the measurement vector z to estimate the initial state. If the vehicle is Earth stationary, this will use the nonlinear gyrocompassing routine that was described in Section 2.1.6. As a byproduct, this block also outputs a pure inertial solution.

The Extended Kalman Filter performs the state estimate. Its inputs are the control input u, measurements z, and the initial state x_o. It does not begin execution until the Initialization block has determined the initial state. The Extended Kalman Filter outputs the state estimate $\hat{x}$ in addition to the measurement error $\tilde{z}$, the estimated covariance P, and the chi-squared statistic χ^2. The details of this block will be described shortly.

The Post Filter block calculates additional navigation outputs that are not part of the state vector. This includes the conversion to NED and geodetic coordinates. Euler angles are also computed in this block. This technique is an alternative to using these quantities as state variables. The goal here is to make the Extended Kalman Filter as simple as possible.

The details of the Extend Kalman Filter itself are shown in Figure 3-3. These four blocks perform the state estimation and linearization and the calculation of the Kalman gain and covariance.

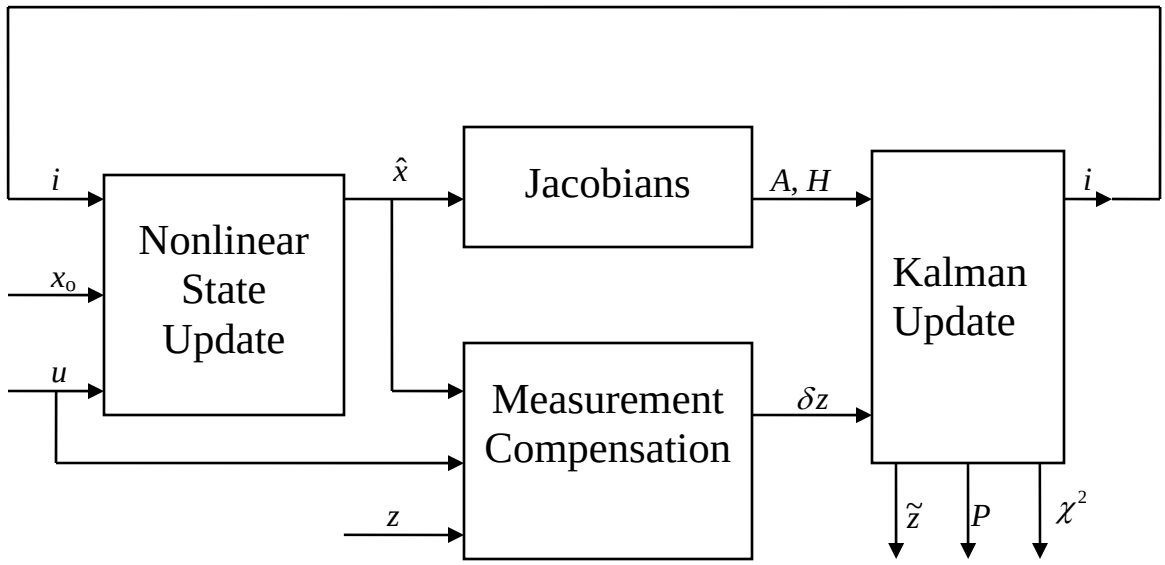

Figure 3-3: Data flow in the Extended Kalman Filter

The Nonlinear State Update block is responsible for propagating and updating the estimate of the full state vector. This requires integrating the full nonlinear state equations, using the initial condition x_o and control input u. It also requires processing the innovations i, which the Kalman filter generates with each measurement.

The Jacobians block linearizes the system and produces the instantaneous state matrices. This is primarily A and H, but can also include F and L. These matrices are calculated using the closed-form solution for the Jacobian that was presented in Section 2.3.2. In this fashion, the differentiation has been performed analytically and the Jacobians are expressed as a function of the current state estimate $\hat{x}$.

The Measurement Compensation block uses the state estimate $\hat{x}$ and control input u to compensate the nonlinear measurement z. This produces a small measurement error δz, where $\delta z = H\delta x$. This allows for a simpler and more accurate linearization of the measurements. This again uses the equations given in Section 2.3.2.

The Kalman Update block is responsible for implementing the Kalman filter, as described in Section 2.2. This includes the propagation of the covariance and the calculation of the Kalman gain. The linearized state matrices are input. The measurement error δz is also input to calculate the innovations i. This filter also requires noise parameters, including the covariance of the initial estimate, the process noise, and the measurement noise. This block uses the continuous-discrete formulation with a constant forgetting factor.

Further details of the system are straightforward matters of programming the equations presenting in the Theoretical Derivation of this report (Section 2). Some care must be taken to organize the order of execution so that linearization and measurement updates use the most recent state estimate.

3.3 Covariance Analysis

The Kalman Filter propagates the covariance of the estimation error for the purposes of calculating the optimal gain. Conveniently, this covariance gives a theoretical prediction of filter accuracy. Covariance analysis can also be used to determine the sensitivity of the filter to different noise parameters. This is important for sensor design and cost tradeoffs, and helps build an understanding of the system observability.

Since covariance analysis is based on the theoretical system model, modeling errors will affect the true accuracy. The analysis of modeling errors that was conducted in Section 2.2.4.3 showed that the true accuracy will be worse than the estimated accuracy because the modeling errors present an additional source of error. In most regards, the predictions of covariance analysis represent the best that the filter can be expected to perform. To further improve performance, the system and sensor noise characteristics must be improved. If performance is significantly worse than covariance predictions, there is probably an error in the model or filter implementation.

For this study, the filter is simulated and the covariance is recorded. To speed up simulation, the state estimate and measurement errors do not need to be calculated. The covariance will typically reach a steady state, although there is a dependency on the flight path. This analysis is conducted using three different flight paths. A large number of parameters are varied through the study, with special focus on sensor noises.

Table 3-1: Nominal Simulation Values

Parameter	Value
Gyro Accuracy	1.64×10^{-2} deg/s
Accelerometer Accuracy	0.0305 m/s^2
GPS Position Accuracy	1 m
GPS Velocity Accuracy	0.1 m/s
GPS Rate	1 Hz
Force Disturbance	1 m/s^2
Torque Disturbance	1 rad/s^2
Other Process Noise	0
Latitude*	45°
Forgetting Rate	0
Inertia (normalized)	1 kg-m^2
Initial Attitude Standard Deviation	3.16 deg
Initial Position Standard Deviation	1 m
Duration	15 min

* Flight data uses the actual latitude as nominal

The nominal simulation, which uses noise and parameters as expected in the actual hardware, is used as a baseline for comparison. The nominal parameters are shown in Table 3-1. Force and torque disturbances have been chosen to be sufficiently unpredictable so that

measurement error dominates the accuracy. The remaining process noise terms, such as in $\dot{p} = v$, were set to zero.

Simulations have been run both with and without biases in order to study their effect. When biases are used, the nominal values are as given by Table 3-2.

Table 3-2: Nominal Bias Values

Parameter	Value
Gyroscope bias	2.78×10^{-4} deg/s
Accelerometer bias	3×10^{-4} g
Bias time constant	20 min

The following sections present covariance analysis using the Earth stationary, aircraft, and booster rocket trajectories. For each trajectory, a number of sensitivities are explored. First, the sensitivity to IMU and GPS noise is studied. Then biases are added and the effect of varying bias levels is explored. For the Earth stationary trajectory, the fixed-to-Earth measurement and the effect of latitude are characterized. For all trajectories, the effects of the GPS rate and the forgetting rate are also studied.

3.3.1 Earth Stationary

The first flight path is Earth stationary, representing a vehicle that is nearly motionless on the Earth's surface. This flight path provides a near worst case, in which the navigation solution does not benefit from vehicle motion. (The case of a vehicle that is stationary in inertial coordinates is worse, but is somewhat unlikely.) Most simulations use only GPS and IMU measurements, but the effect of the fixed-to-Earth measurement is also explored.

Each simulation produces a time plot of the covariance of the attitude and position estimation errors. For analysis, it is convenient to resolve these 3 by 3 covariance matrices into their principal values and to express the result as a standard deviation. The nominal simulation is shown in Figure 3-4. It can be seen that accuracy improves asymptotically as time progresses and approaches a steady state value.

The position accuracy dynamics are relatively fast because position is measured directly by the GPS. The attitude plot shows both a fast and slow response. The direction of the slow response corresponds to rotation about the local vertical, commonly known as heading (NED yaw). This plot suggests that heading accuracy continues to decrease after the 15 minute simulation, but 15 minutes was chosen as a reasonable time for comparison.

The substantial difference between heading accuracy and other attitude accuracy is the result of the sensor architecture. This behavior is in agreement with the observability analysis in Section 2.3.3 and the discussion of initial alignment in Section 2.1.6. Strong attitude alignment is possible by aligning the force vector, which, in this case, is the reaction to the

gravitational force. As the Earth rotates, the direction of this force slowly changes, which allows for weak heading alignment.

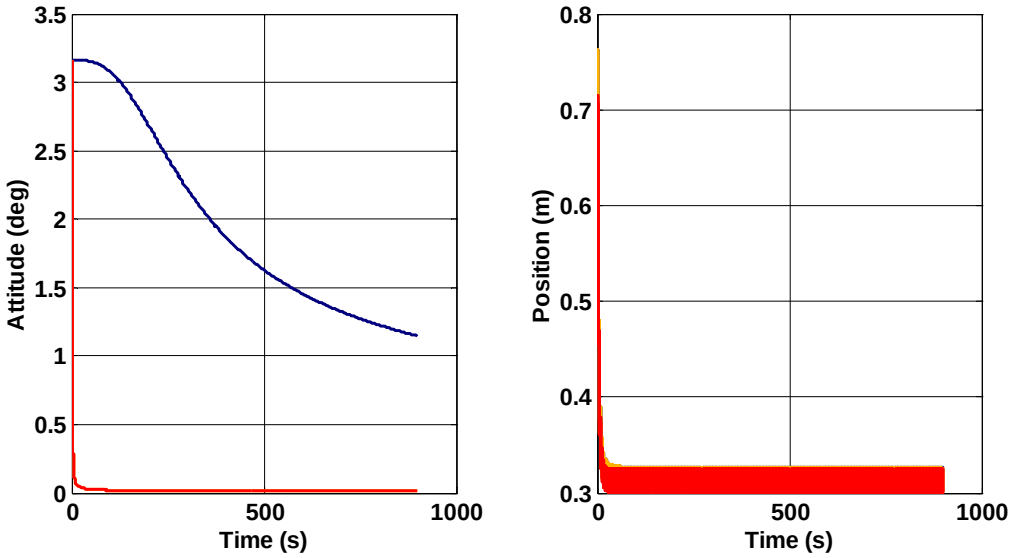

Figure 3-4: Time Plot of Nominal Attitude and Position Accuracy (principal standard deviations)

Throughout the analysis, attitude exhibits this characteristic of having one dimension of poorer accuracy. It is frequently aligned with heading, because the force vector tends to be in the vertical direction. For most of the analysis, we will plot this worst dimension separately from the other dimensions.

A set of simulations have been constructed to explore the effects of varying parameters. The final accuracy after 15 minutes is used for comparison between simulations. The accuracy results are then sorted by the varied parameter.

Figure 3-5, shows the first of these simulations. In this case, the effect of varying the measurement noise of the IMU and GPS is studied. The figure is divided into three panes which show the accuracy in the worst attitude direction (left plot), other attitude directions (middle plot), and position (right plot). This plot format is used throughout the covariance analysis.

For each plot, the accuracy is plotted against the relative noise on a logarithmic, where 10^0 represents the nominal noise value given by Table 3-1. These plots are interpreted by looking at the slope of each curve. Each line represents the effect of varying a different parameter. A steep curve means that accuracy is sensitive to that parameter, and the predicted accuracy values can be read from the vertical axis.

Beginning at the left, the worst attitude is most sensitive to the gyroscope accuracy. It is also sensitive to accelerometer accuracy. These dependencies exist because the IMU is used to align the attitude. This direction requires detection of Earth rate, so the gyroscope noise is critical. The sensitivity to GPS position and velocity is very small.

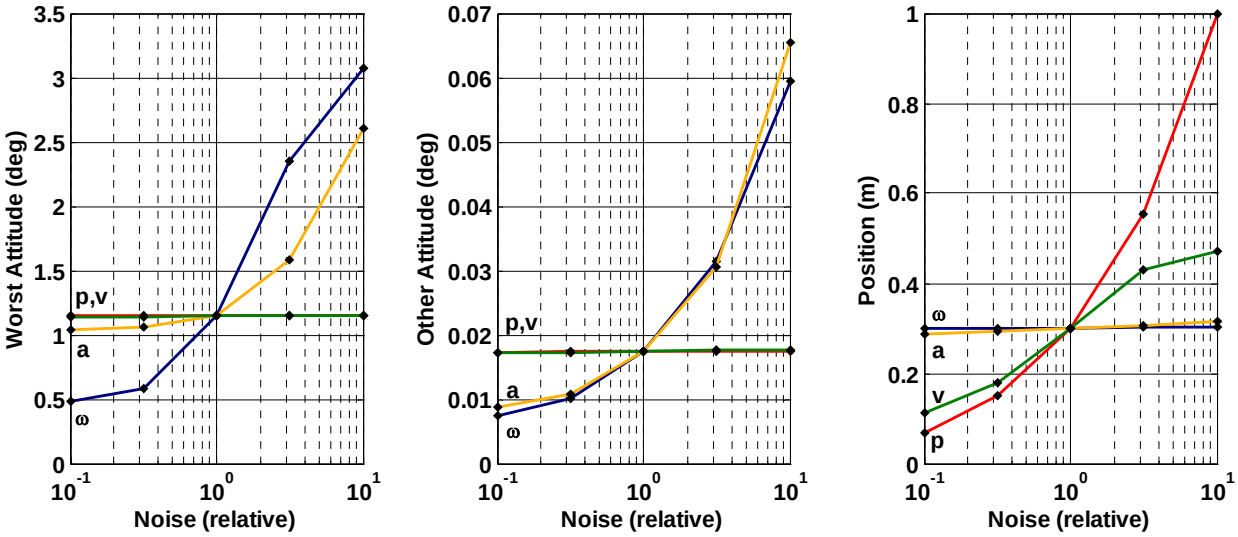

Figure 3-5: Accuracy under Varying Noise in the Gyroscopes (ω), Accelerometers (a), GPS Position (p), and GPS Velocity (v)

The other attitude directions (middle plot) have equal dependence on gyroscopes and accelerometers. These attitude dimensions can be oriented using gravity vector alignment.

Position accuracy (right plot) depends on the accuracy of the GPS position and velocity. The accelerometer noise has a small effect on accuracy.

The difficulties of heading alignment can be dramatically reduced if it is known that the vehicle is stationary with respect to the Earth. This is treated by the filter as an additional measurement of body rate and velocity. The accuracy of this measurement is the amount of vehicle motion or vibration that is present when this measurement is used. Nominal values for this accuracy are shown in Table 3-3.

Table 3-3: Nominal Values for Fixed-to-Earth Measurement

Parameter	Value
Rate noise	0.01 deg/s
Velocity noise	1 mm/s

Figure 3-6 shows the effect of varying the amount of motion relative to these values. At these levels, the fixed-to-Earth measurement greatly improves accuracy. It is therefore highly recommended that this feature be used when the vehicle is Earth stationary. For comparison, the original result, where the vehicle was stationary but the fixed-to-Earth measurement was not used, is plotted as the point labeled “not fixed”.

The increase in attitude accuracy is the result of an improved measurement of body rates, as is evident from the steep slopes in the accuracy plot. When the vehicle is Earth stationary, the body rotation rate is equal to the rate of Earth rotation, which is precisely known. There

remains some uncertainty, because the vehicle may not be absolution stationary, but this uncertainty can be less than the gyroscope noise.

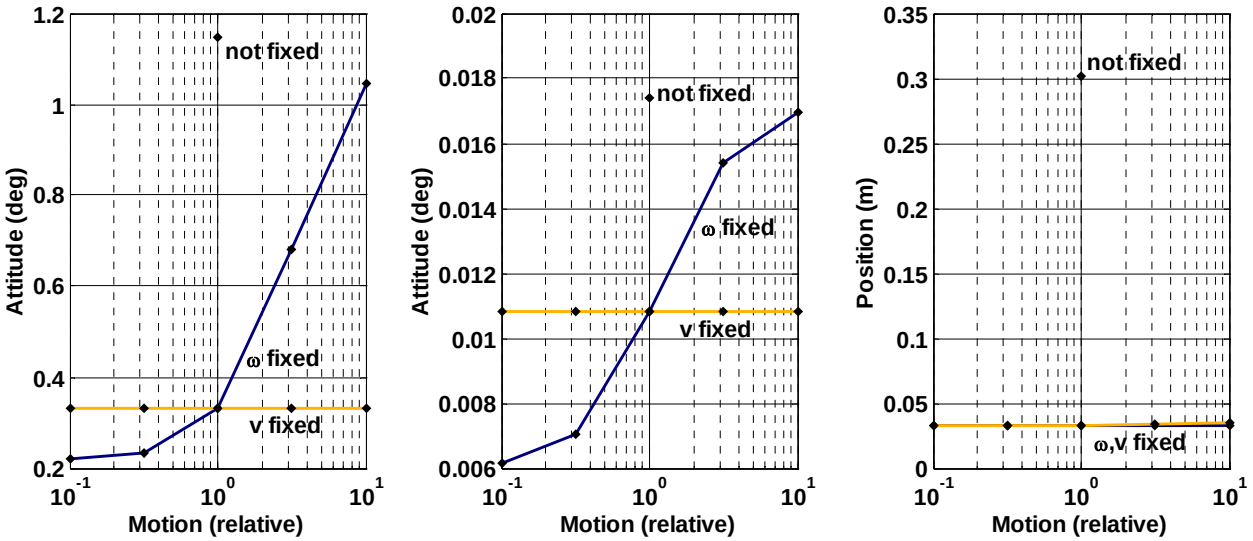

Figure 3-6: Accuracy using Fixed-to-Earth Measurement

Position accuracy is also greatly improved. Without the fixed-to-Earth measurement, position accuracy stems from the accuracy of the GPS and the accelerometers. With this measurement, the velocity is precisely known. As a result, the noise in the accelerometers is no longer harmful to the position estimate. The position accuracy curves are flat for these noise values because the fixed-to-Earth velocity measurement is much better than the accelerometers. However, the fixed velocity curve will eventually slop upward if the vehicle vibrations increase to levels comparable to the accelerometer noise.

A study of the effect of biases in the gyroscopes and accelerometers has also been conducted, using the nominal bias parameters listed in Table 3-2. The effect of biases when using the fixed-to-Earth measurement is shown in Figure 3-7. In this case, the gyroscope bias is fully observable, and only the accelerometer bias has an effect on attitude accuracy. The position accuracy is not affected, because the GPS measurements remain unbiased.

The effect of biases when the fixed-to-Earth measurement is not used is shown in Figure 3-8. In this case, the attitude performance is significantly worse. The heading error is now dependent primarily on the magnitude of the gyroscope bias. This is because both the heading error and the gyroscope heading bias must now be observed through Earth rotation.

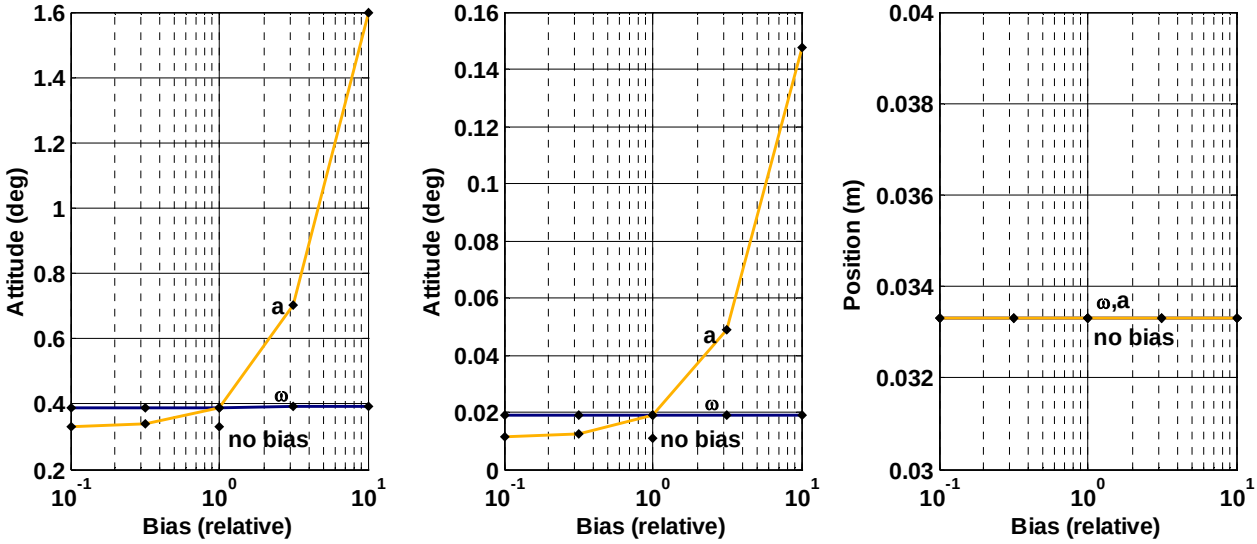

Figure 3-7: Accuracy with Biases using the Fixed-to-Earth Measurement

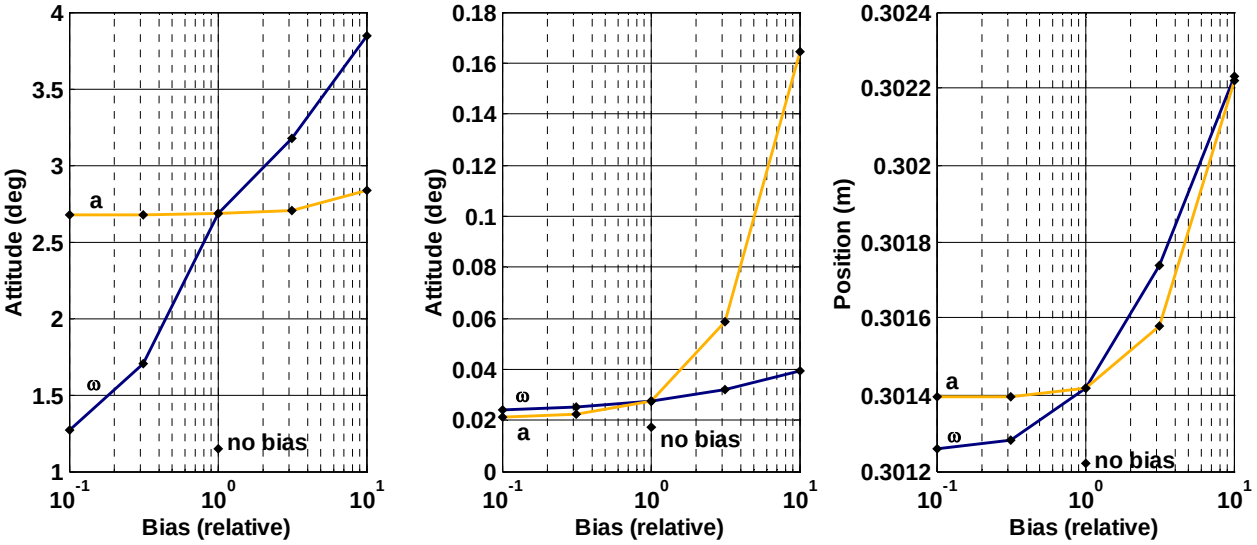

Figure 3-8: Accuracy with Biases without using the Fixed-to-Earth Measurement

The effect of the latitude variations is shown in Figure 3-9. Locations near the Earth's poles suffer because the Earth rotation vector is aligned with the gravity vector, so the applied force remains constant as the Earth rotates. Note that the initial accuracy for these simulations was 3.16°, so there is no alignment improvement at the pole.

The effect of increasing the GPS rate is shown in Figure 3-10. Attitude accuracy is not significantly affected, but positioning accuracy increased. This is the result of averaging more data points. With N samples, the standard deviation reduces by $\sqrt{N}$. However, this result relies on the assumption that the GPS noise is independent of previous values.

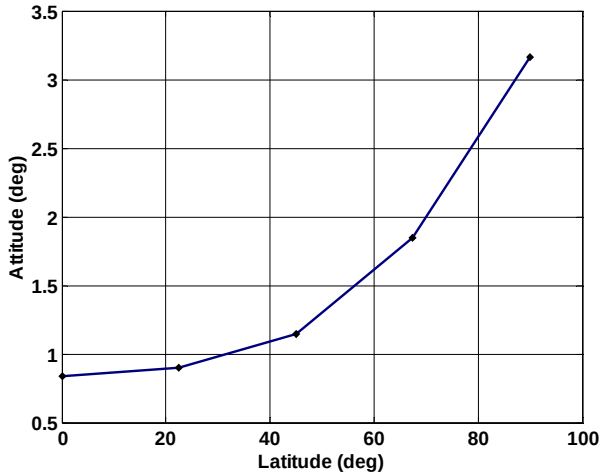

Figure 3-9: Effect of Latitude on Attitude Accuracy

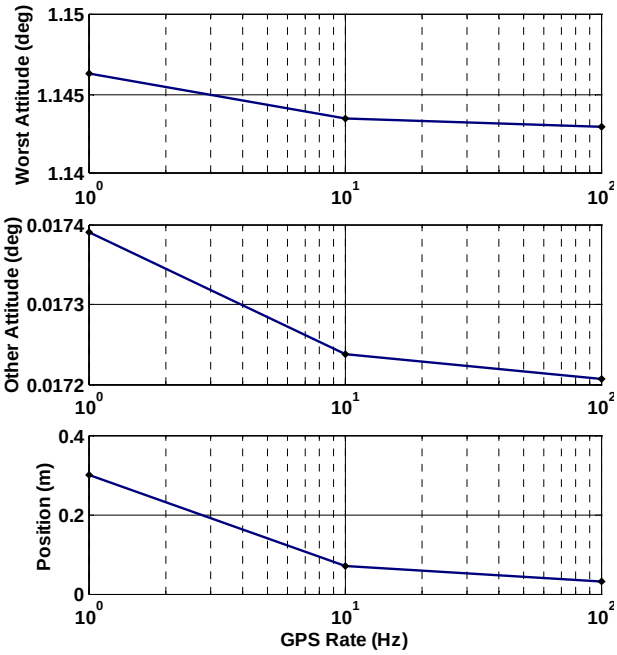

Figure 3-10: Effect the GPS Rate

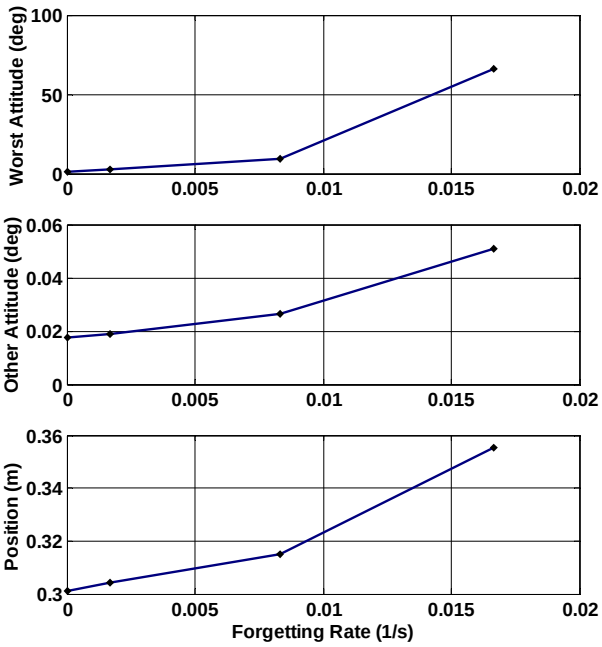

Figure 3-11: Effect of the Forgetting Rate

Finally, the effect of data forgetting is shown in Figure 3-11. A faster forgetting rate produces larger covariance, because less data is used in the solution. This result is misleading though, because the covariance has been modified to produce forgetting. The benefit of data forgetting is robustness, since poor measurements and filter glitches are more quickly ignored. In fact, a large variance will lead to higher gains in the Kalman filter which will cause faster convergence in the actual error. Primarily, we must ensure that the covariance remains reasonable for our model. The worst attitude covariance at large forgetting rates begins to stretch the small angle assumption.

3.3.2 Aircraft Flight Path

Simulations were repeated using the aircraft flight data, which contains measurements from sensors aboard a small airplane. The results demonstrate that flight path complexity can greatly increase the attitude accuracy. More details of this flight path are shown later in the full simulations of Section 3.5.1.

The time plot of accuracy for the nominal simulation is shown in Figure 3-12. The attitude accuracy has improved significantly, as a result of the richness of the flight path. Takeoff, landing, and aircraft maneuvers all generate large variation in the direction of applied force. Heading accuracy was poor prior to takeoff.

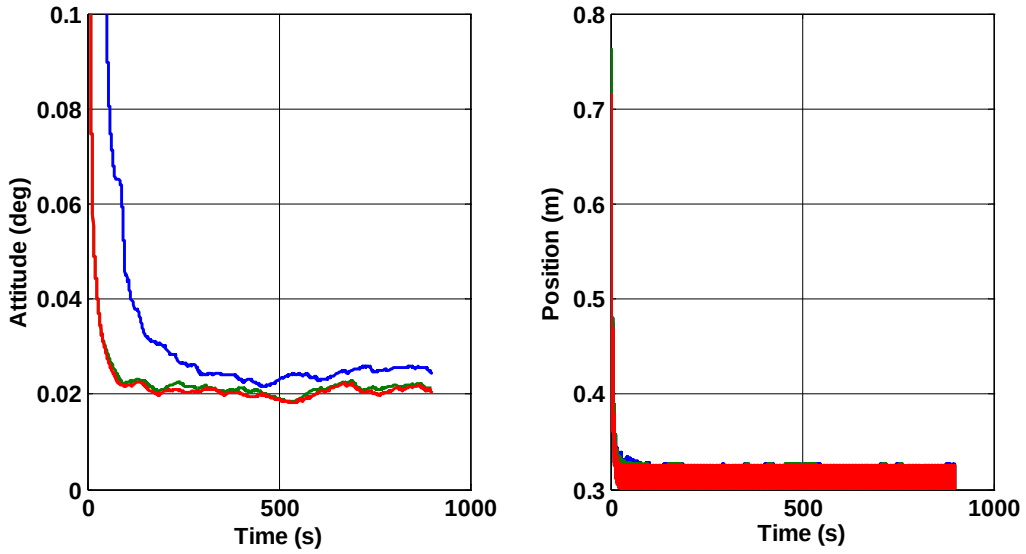

Figure 3-12: Time plot of Nominal Attitude and Position Accuracy

The sensitivity to individual parameters can be studied again using this flight path. Figure 3-13 shows the effect of varying the measurement noise of the IMU and GPS. Attitude accuracy is very good and all three dimensions show similar accuracy, due to the substantial variation in the direction of force. Attitude accuracy depends equally on the gyroscopes and accelerometers. Gyroscope measurement of Earth rate is extremely difficult in a moving vehicle, but accurate gyroscopes still help to connect regions with different force directions.

The position accuracy is the same as for Earth stationary operation.

The effect of biases in the IMU is shown in Figure 3-14. Bias compensation is very successful and the estimate is almost as good as the estimate without biases. This is because the biases are highly observable under this flight path.

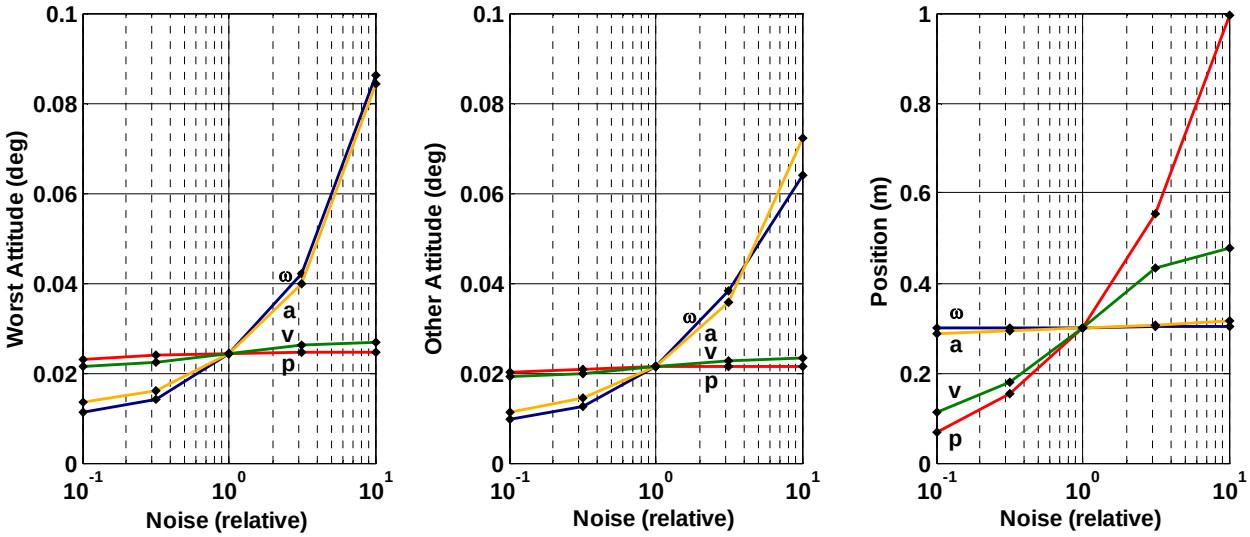

Figure 3-13: Accuracy under Varying Noise in the Gyroscopes (ω), Accelerometers (a), GPS Position (p), and GPS Velocity (v)

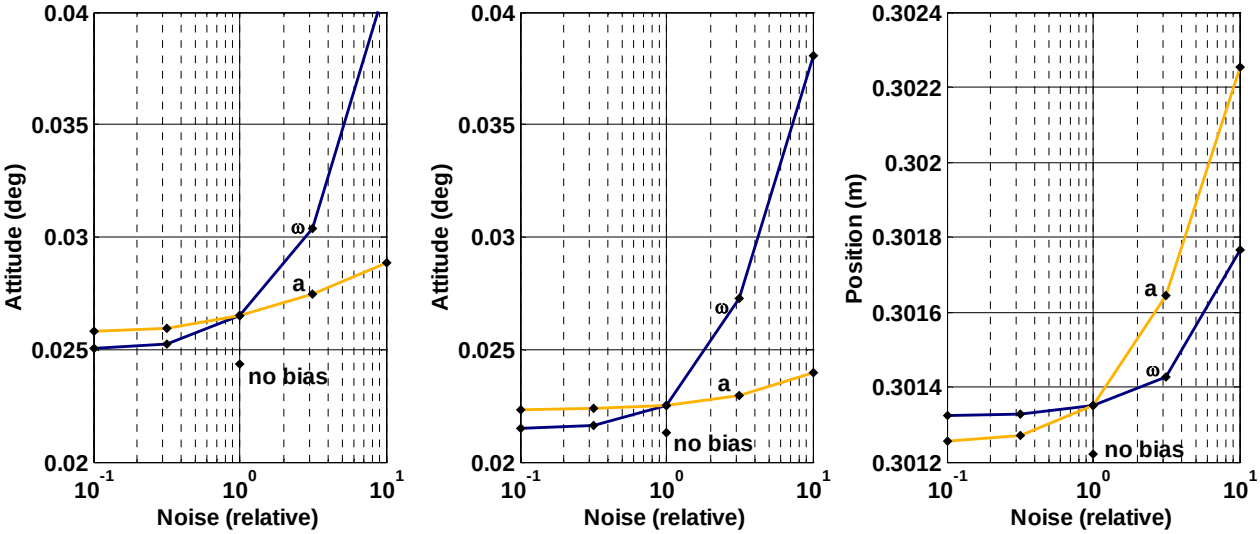

Figure 3-14: Accuracy under Varying Gyroscope Bias (ω) and Accelerometer Bias (a)

The effect of the GPS Rate is shown in Figure 3-15. There is slight improvement at faster GPS rates. Remember that this is only true when the GPS errors of successive measurements are independent.

The effect of data forgetting is shown in Figure 3-16. In this flight path, forgetting at faster rates is quite acceptable. It may be advantageous to use a higher forgetting rate to provide robustness to anomalous conditions.

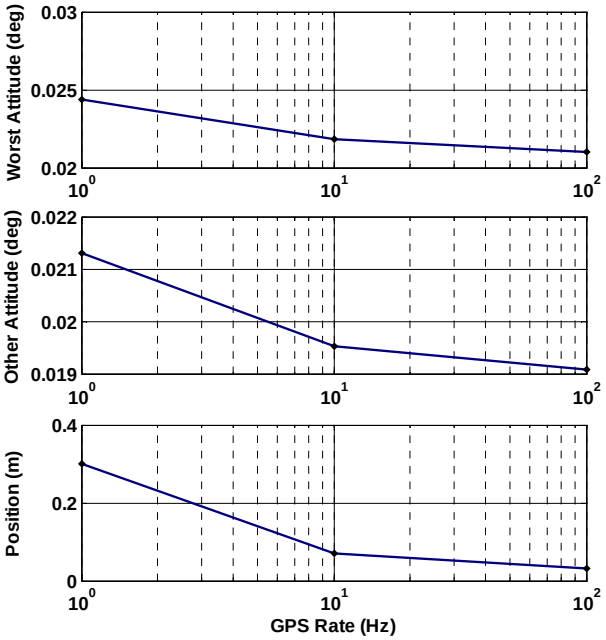

Figure 3-15: Effect of the GPS Rate

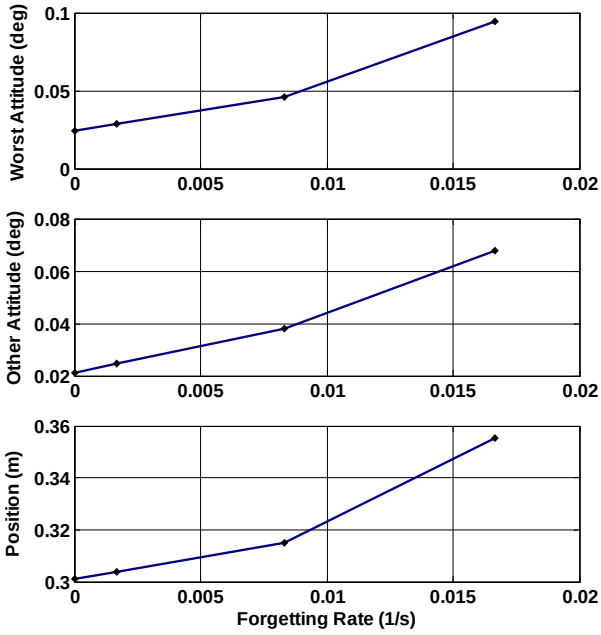

Figure 3-16: Effect of Data Forgetting

3.3.3 Booster Flight Path

The booster flight data represents rocket-powered flight. This dataset contains three distinct operating modes. The data begins before liftoff, when the rocket is Earth stationary. This is followed by the boost phase, which contains extremely large accelerations and disturbances. Force here is predominantly in the axial direction, but the rocket trajectory curves slowly. Following the boost phase is a period of exoatmospheric maneuvering, which contains very little measured acceleration because the vehicle is in freefall. With the high speeds of a rocket, this flight path covers very large distances. More details of this flight path are shown in the full simulations of Section 3.5.2.

The simulation time plots of attitude and position accuracy are shown in Figure 3-17. A large improvement in the worst-direction attitude accuracy occurs at liftoff ($t = 0$) and during the boost phase where the force is large. The fact that attitude accuracy began improving before liftoff is the result of estimated ground vibrations, and will be discussed with full simulation in Section 3.4.3. In the final portion of this data, corresponding to exoatmospheric flight, the attitude accuracy degrades slightly because the measured acceleration is small. (This effect is barely visible on these plots, but is readily apparent at smaller scales.)

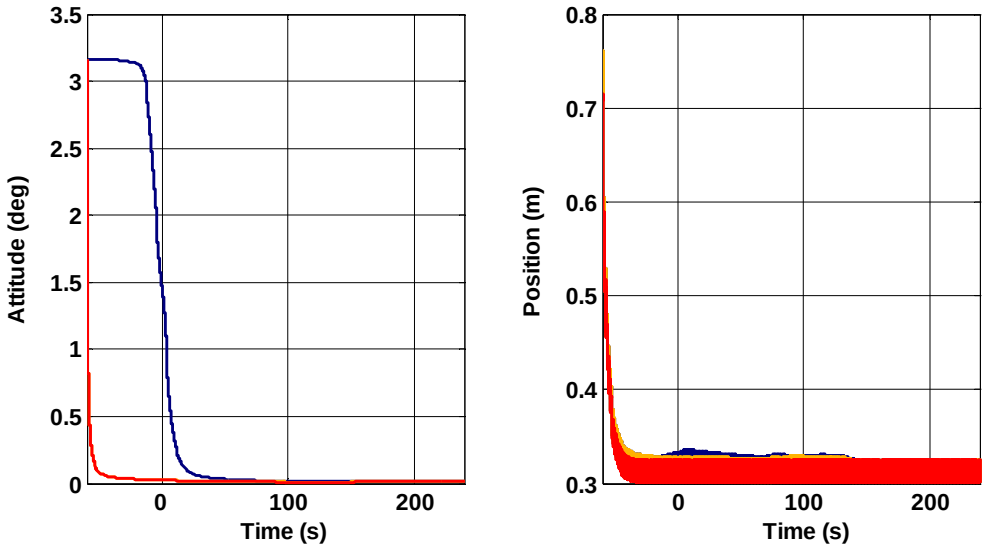

Figure 3-17: Time Plot of Nominal Attitude and Position Accuracy

Accuracy values under varying noise conditions are plotted in Figure 3-18. Accuracy is very similar to the accuracy using the aircraft trajectory, although slightly poorer in the worst attitude direction. This corresponds to axial rotation (roll), because the boost acceleration is almost entirely in the axial direction. Note that attitude accuracy shows almost no sensitivity to GPS accuracy. This is because distances and velocities are very large in this trajectory, so small amounts of GPS noise are irrelevant for attitude alignment.

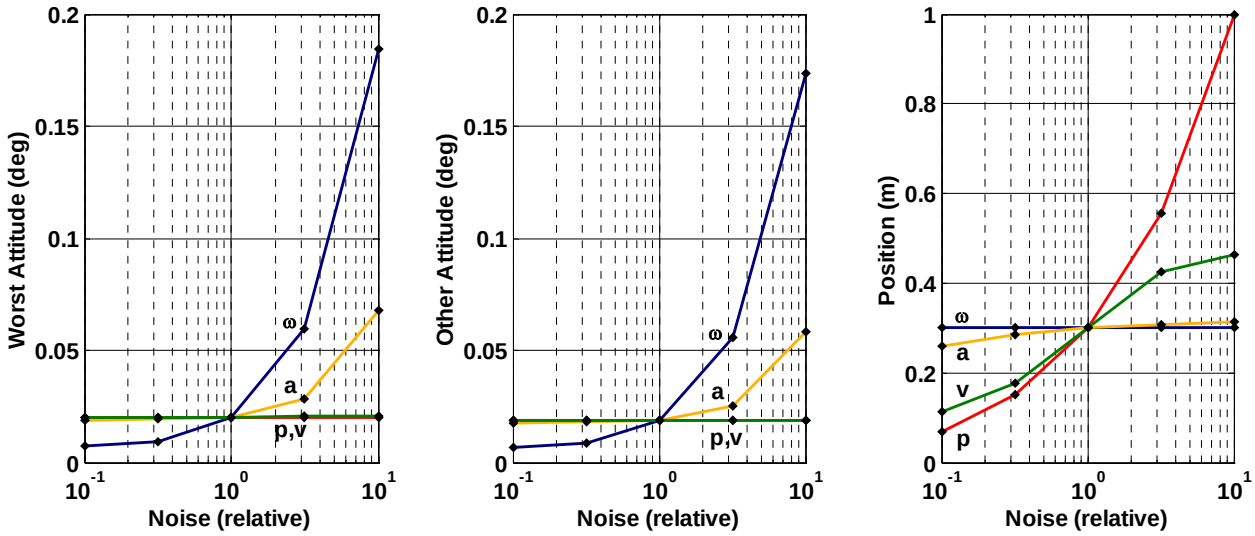

Figure 3-18: Accuracy under Varying Noise in the Gyroscopes (ω), Accelerometers (a), GPS Position (p), and GPS velocity (v)

The effect of biases is shown in Figure 3-19. Once again, the trajectory is rich enough to observe these biases, and they do not seriously affect accuracy. The booster trajectory shows more sensitivity to gyroscope biases and less sensitivity to accelerometer biases, in comparison with the aircraft trajectory.

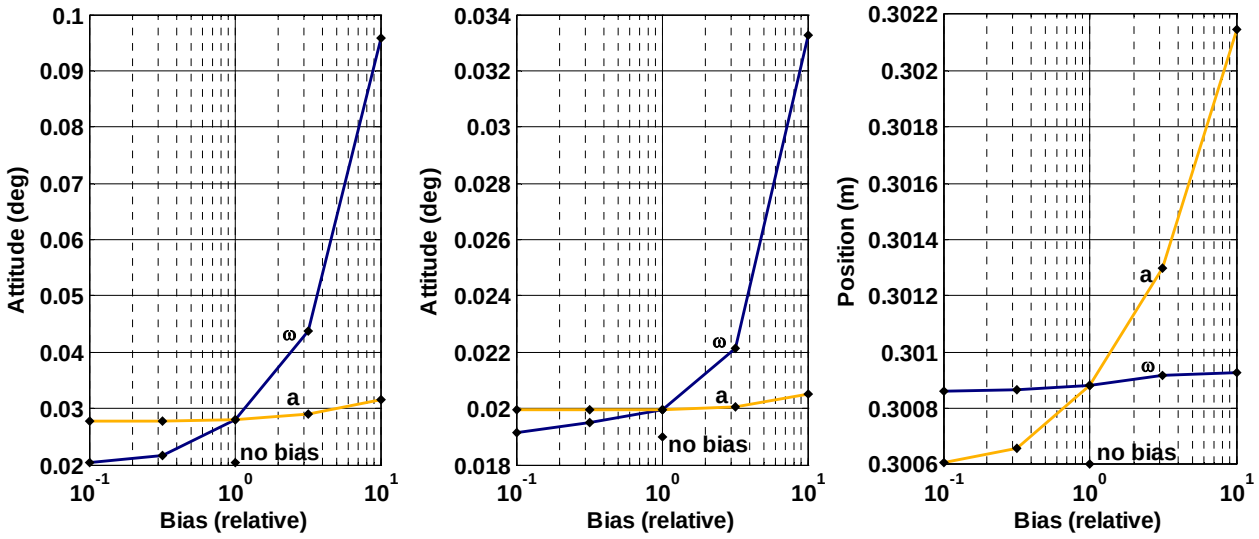

Figure 3-19: Accuracy under Varying Gyroscope Bias (ω) and Accelerometer Bias (a)

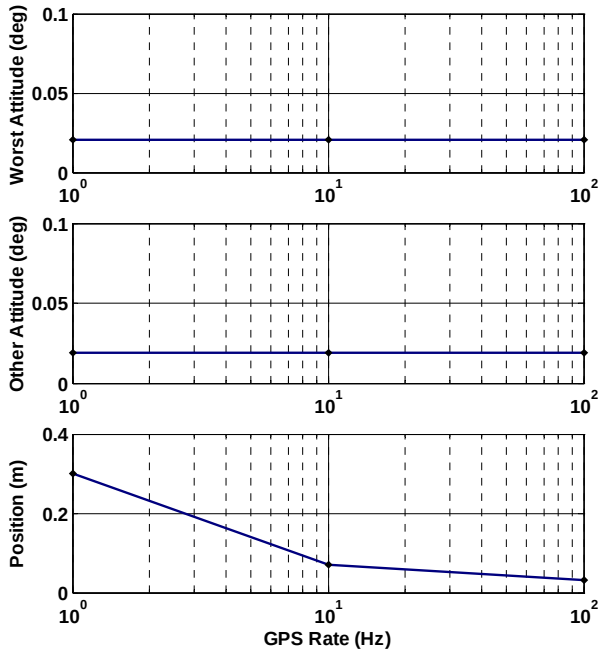

Figure 3-20: Effect of GPS Rate

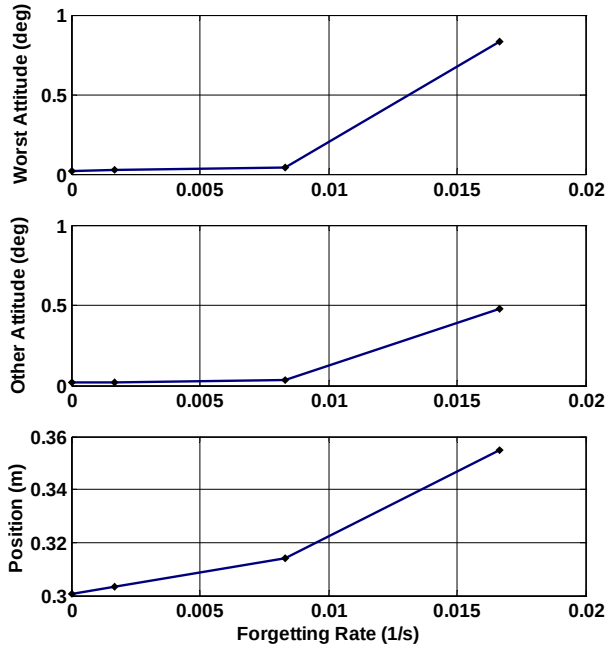

Figure 3-21: Effect of Data Forgetting

Figure 3-20 shows the effect of the GPS rate, which does not affect attitude accuracy in the booster trajectory. This again is a reflection of the reduced sensitivity to GPS accuracy because large distances are traveled.

The effect of the forgetting rate is shown in Figure 3-21. Attitude covariance under forgetting is an order of magnitude larger than the covariance using the aircraft trajectory. The booster trajectory richness is due to the gradual arc in the flight path, which is only visible over long periods of time. Also, the booster trajectory richness is concentrated in the boost phase of flight.

3.4 Simulated Data

The full performance of the filter has also been characterized in simulation. The following analysis expands on the previous covariance analysis by introducing simulated measurements. These measurements are created from a known truth model, but are corrupted by noise using pseudorandom number generators. The state estimate is calculated and the filter performance is studied under highly-controlled noise conditions.

The primary output of these simulations is a time plot of the estimation error. For comparison, the estimated standard deviation of the filter error is also plotted. This is the same as the covariance analysis conducted previously and represents an expected convergence curve. The actual error is a random variable, but should have statistics that match the estimated standard deviation.

A variety of studies have been performed in this context. Section 3.4.1 studies normal filter operation. In these conditions, the filter is given the correct noise parameters. As a result, the initial errors and measurement noise are consistent with the noise model of the filter. The process noise for disturbances and bias dynamics is also consistent. A rich trajectory is used, so that there is ample information to determine all state estimates.

Section 3.4.2 studies filter robustness to anomalous conditions, where the filter model is no longer consistent with the measurement data. Here, the filter uses underestimated values for initial errors, measurement noise, and process noise. Measurement glitches are also studied. Note that overestimated noise does not present a problem in filter operation. It does cause suboptimality, however, because the filter discredits valid information.

Section 3.4.3 studies behavior under other flight paths, including the Earth stationary, aircraft, and booster rocket trajectory. The latter two datasets are derived from the flight data, but have been modified to provide an exact truth model and adjustable noise parameters. In effect, a simulated dataset was created with a trajectory that closely matches the flight data. These simulations provide analysis of observability and accuracy in realistic flight environments.

Finally, section 3.4.4 analyzes the effect of removing bias states from the filter. This is motivated because these additional states add computation. If they do not produce improved accuracy, they should be removed.

3.4.1 Nominal Operation with Initial Errors

The first test of filter performance is under normal operating conditions. These simulations use a correct noise model so that the actual noise values match the filter's covariance parameters for measurement, process, and initial state. The state estimate is initialized with an error of one standard deviation in each element of the state vector.

For these simulations, a rich trajectory is used, shown in Figure 3-22. This trajectory contains gradual sinusoidal variations in attitude and rapid variations in the body force. The torque that is used to produce this attitude variation is also plotted. A full truth data set, including position, velocity, and body rates, is created from this force and torque. The force and torque of this trajectory are given to the filter as a control input, but the filter is required to estimate additional force and torque disturbances. This trajectory is not intended to represent realistic motion, but provides good characterization of the filter behavior in situations with rich information. The effect of other trajectories is pursued later in Section 3.4.3.

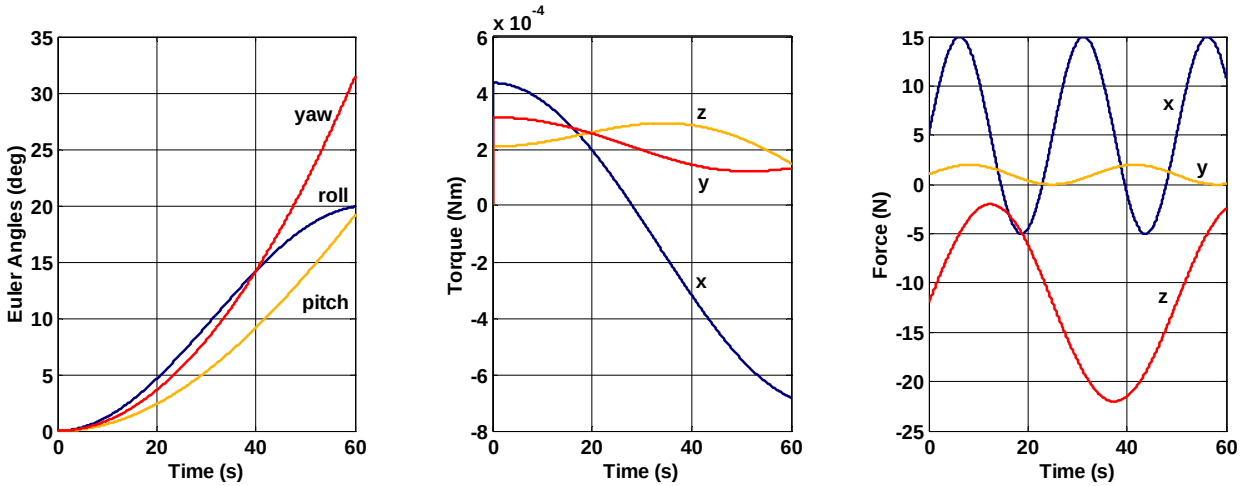

Figure 3-22: Trajectory for Normal Operation Simulations

The initial errors, which matched the initial state standard deviations, are listed in Table 3-4. These values have been chosen to be reasonable and consistent with the hardware. However, the actual biases are very small, and produce simulations that are successful but uninformative. The biases are dominated by the white measurement noise, and the effect of biases is obscured. For the purposes of illustration, most studies increase the biases by a factor of 100. This makes proper bias compensation critical for filter performance. In most cases, constant biases are used in simulation.

The attitude response is shown in Figure 3-23. Convergence is good in all three dimensions, due to the rich trajectory. Error remains in good proportion with the estimated standard deviation. Progress in the estimate occurs most noticeably during the 1 Hz GPS updates.

Body rate response is shown in Figure 3-24. Note that the time scale is extremely compacted. Because body rates are measured accurately by the gyroscopes, the rate error is nearly eliminated after the first IMU measurement. Torque disturbance convergence, shown in Figure 3-25, requires two measurements to observe, as the derivative of body rate. There is again good agreement between the estimated standard deviation and the actual error.

Table 3-4: Initial Errors for Nominal Simulation

State	Initial Error
Attitude	3.16°
Body Rate	1°/s
Torque Disturbance	1 rad/s^2
Position	10 m
Velocity	10 m
Force Disturbance	10 m/s^2
Gyroscope Bias*	2.78×10^{-2} deg/s
Accelerometer Bias*	3×10^{-2} g
Gravity Model Bias*	1×10^{-4} m/s^2

* 100 times actual value

Table 3-5: Measurement Noise for Nominal Simulations

Measurement	Noise Standard Deviation
Gyroscopes	0.016 deg/s
Accelerometers	0.30 m/s^2
GPS Position	1 m
GPS Velocity	0.1 m/s

Table 3-6: Distubance Time Constants

Noise	Time Constant
Torque Disturbance	1 s
Force Disturbance	1 s

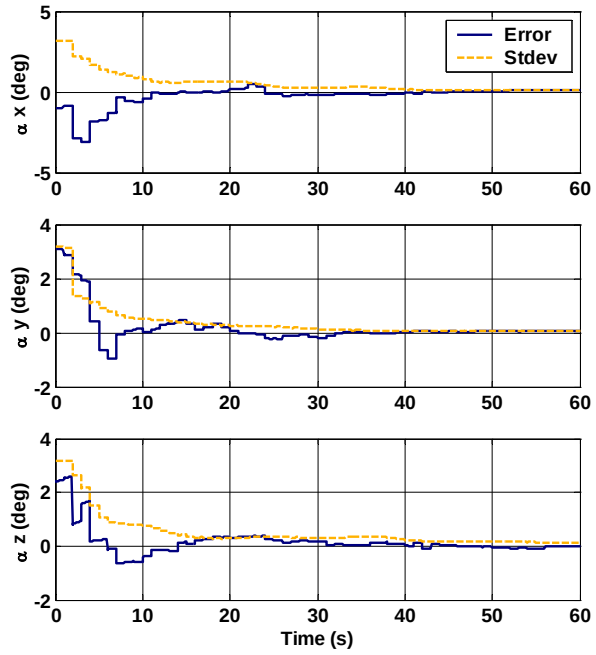

Figure 3-23: Attitude Error Response

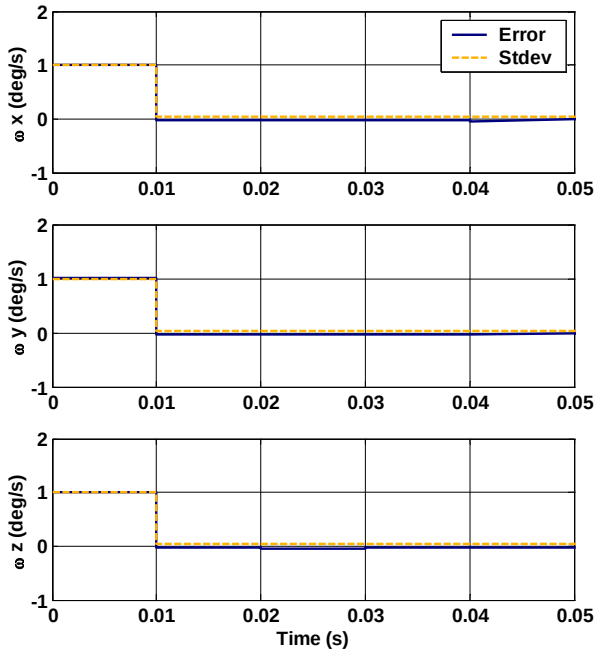

Figure 3-24: Body Rate Response

The position response is shown in Figure 3-26, and shows rapid improvement with the first GPS measurement. The initial increase in error is the result of the velocity error, which is shown in Figure 3-27. The force disturbance response is shown in Figure 3-28. One accelerometer measurement is required to measure this disturbance.

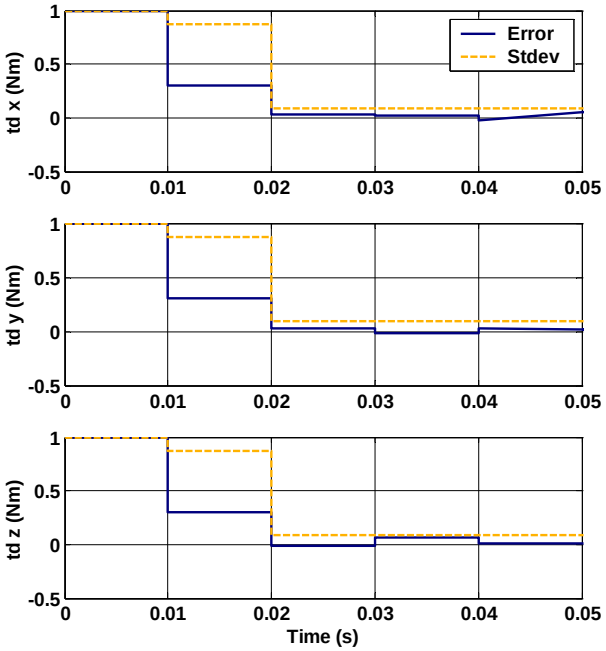

Figure 3-25: Torque Disturbance Response

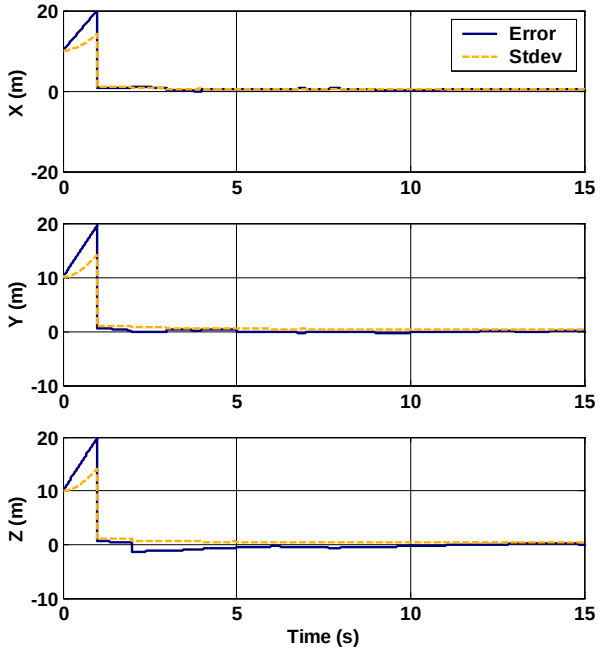

Figure 3-26: Position Response

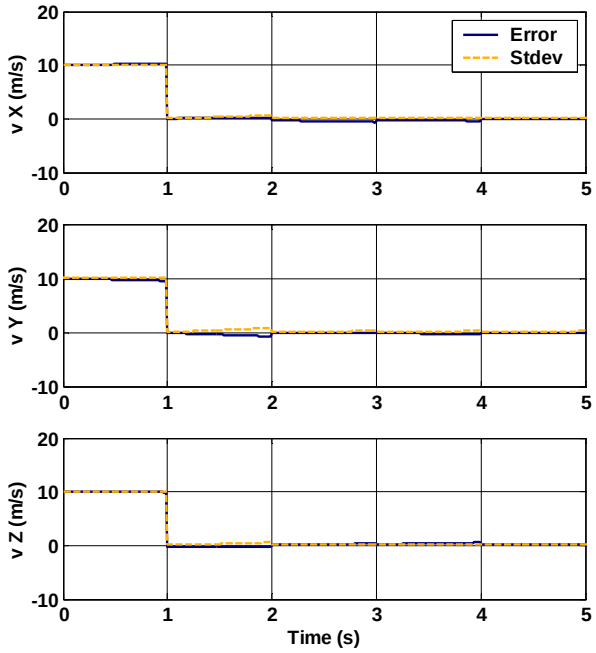

Figure 3-27: Velocity Response

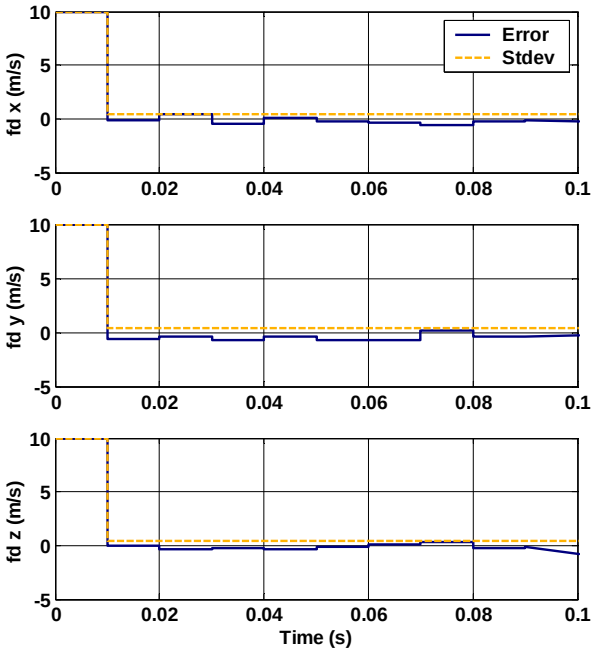

Figure 3-28: Force Disturbance Response

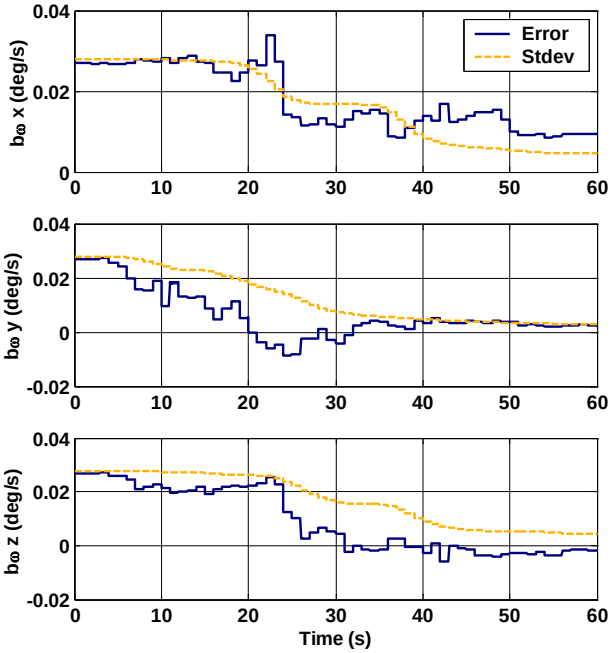

Figure 3-29: Gyroscope Bias Response

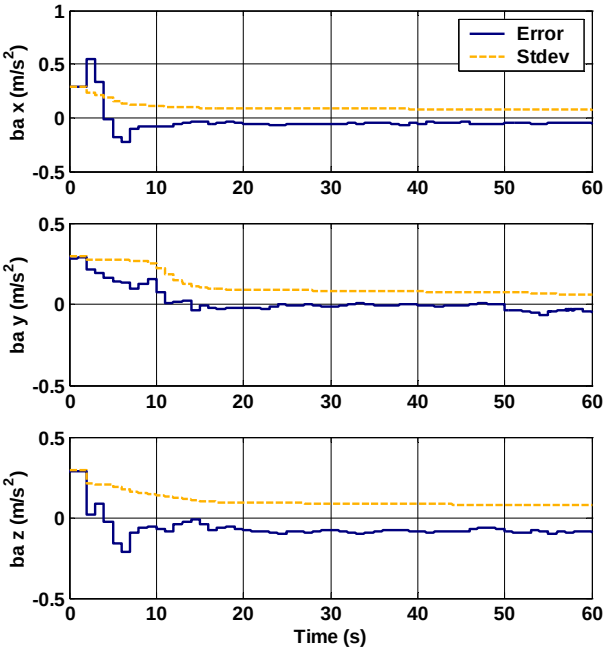

Figure 3-30: Accelerometer Bias Response

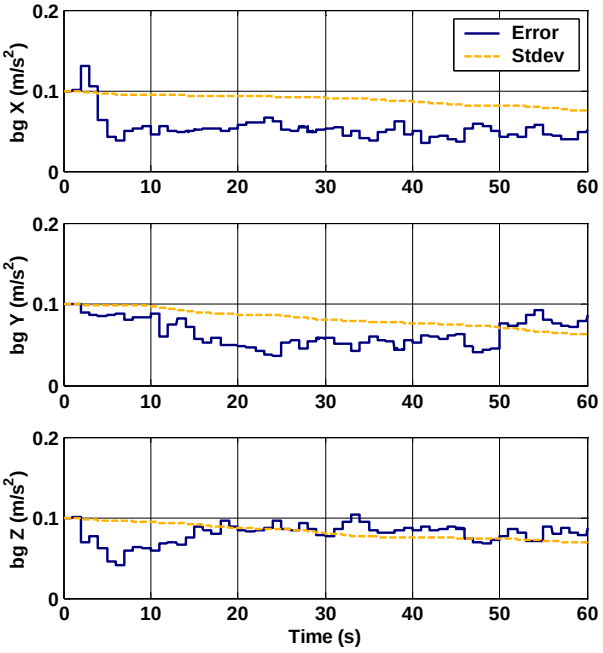

Figure 3-31: Gravity Model Bias Response

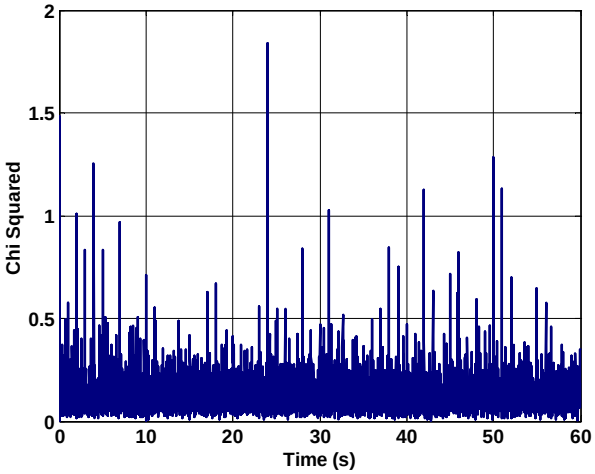

Figure 3-32: Chi-Squared Statistic

The response for biases is shown in Figure 3-29 for gyroscopes, Figure 3-30 for accelerometers, and Figure 3-31 for the gravity model. These biases converge more slowly,

and depend on the richness of the trajectory. It can be seen by the estimate standard deviations that certain periods of time offer information towards these state estimates, while others do not. Ultimately, the convergence of these states is not required because the estimate of position and attitude is of greater importance. However, convergence of the bias states will produce proper bias compensation and increase overall accuracy. It all cases, the estimates of bias state must remain small and bounded.

Finally, the chi-squared statistic is plotted in Figure 3-32. The value remains small, signifying that the measurement errors are in agreement with the estimated covariance. The mean appears to remain constant, which is a sign of healthy filter operation.

3.4.2 Unexpected Noise and Disturbance

These simulations explore abnormal conditions where the filter model is not in full agreement with the measurement data. These are situations that exceed filter design specifications. By studying these conditions, we wish to characterize filter robustness in this regard.

These anomalies fall into several categories. The first set is underestimated noise, where the filter uses noise covariances that are too small. This includes the covariance of the initial state, the measurement noise, and the process noise. The second set of anomalies is measurement glitches. In these simulations, bad measurements are intentionally inserted into the data. This can be either an isolated glitch or a step in the measured value.

Throughout this section, the benefit of data forgetting is explored. In many situations, data forgetting will improve filter recovery by eventually discarding the anomalous information.

Underestimated Initial Errors

We begin with the case of underestimated initial errors. The actual initial errors are maintained at the same levels as in the nominal simulations, but the initial covariance is reduced. This leads to an initial error that is much larger than the estimated standard deviation. As a result, the Kalman filter calculates gains that are too small and the response can be slow.

The case where the initial attitude error is underestimated by a factor of 10 is shown in Figure 3-33. The result shows somewhat erratic behavior, with eventual approximate convergence. Note the steady state error in the *z* direction. The estimates of other variables are generally good, although irregularities in the bias estimates result from attempting to accommodate the data.

This behavior can be dramatically improved using data forgetting. Figure 3-34 repeats the simulation with a forgetting time constant (1/rate) of 30 seconds. This fast forgetting rate causes the estimated standard deviation to grow with time, which keeps gains artificially high and leads to proper convergence. From another viewpoint, this technique forgets the initial condition.

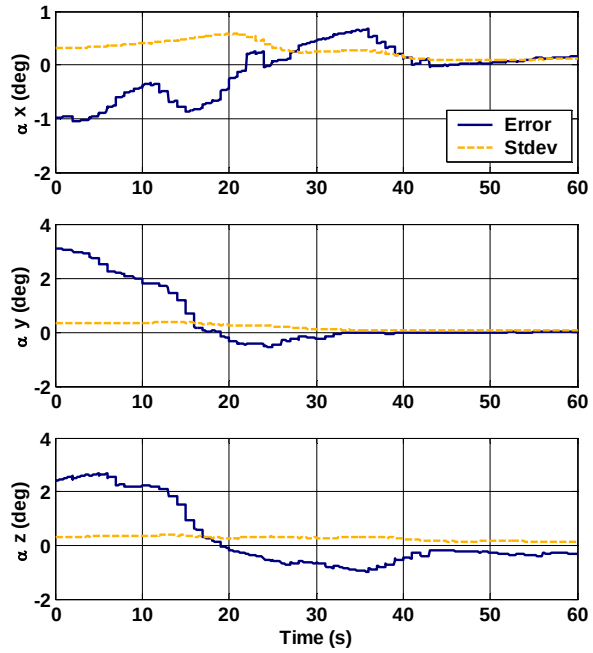

Figure 3-33: 10x Underestimated Attitude Error

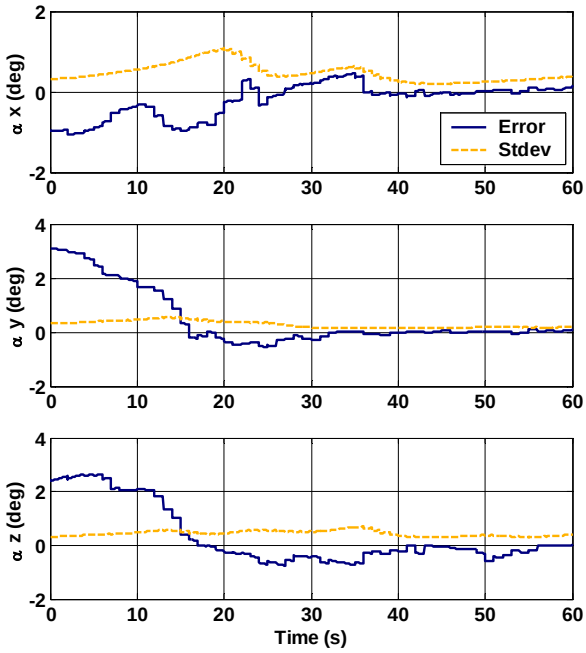

Figure 3-34: Benefit of Forgetting

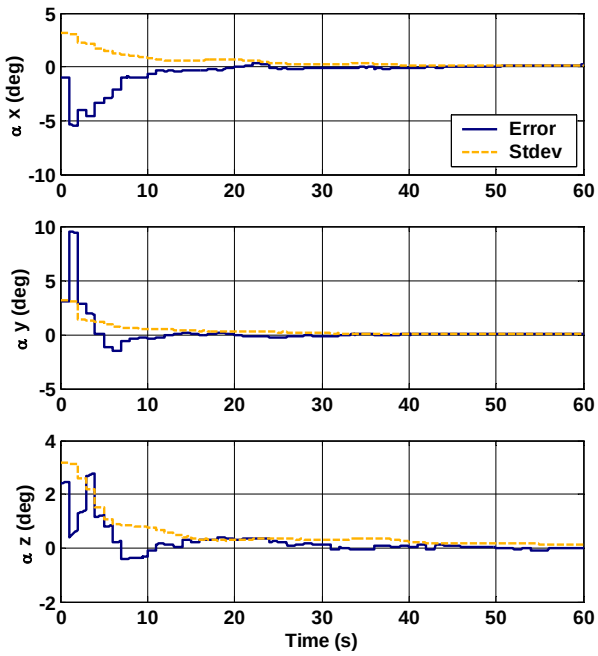

Figure 3-35: Attitude Error caused by Underestimated Position Error

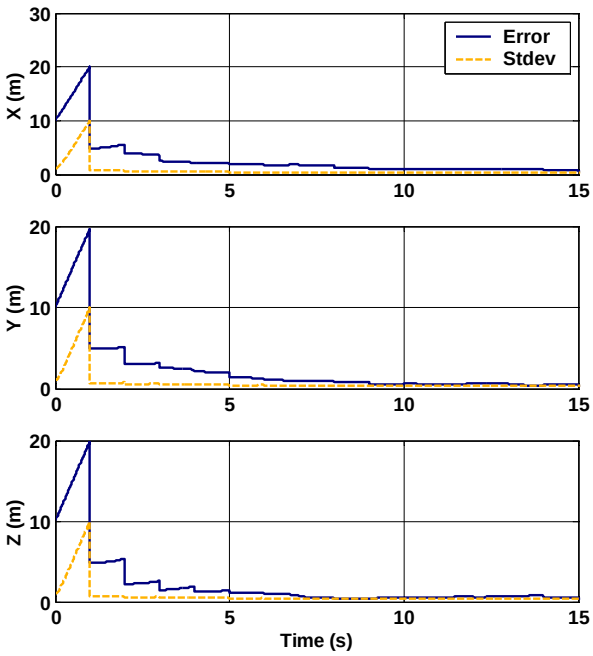

Figure 3-36: 10x Underestimated Position Error

The effect of underestimating the initial position is less serious, because position is directly measurable. An error does result in both attitude (Figure 3-35) and position (Figure 3-36), but these errors diminish with time.

The chi-squared statistic is a good tool for diagnosing this type of problem. The result for the previous simulation of underestimate position is shown in Figure 3-37. The first several GPS measurements create large spikes because they do not match the estimated state. As the state estimate improves, the GPS measurement error becomes smaller and the chi-squared statistic decreases.

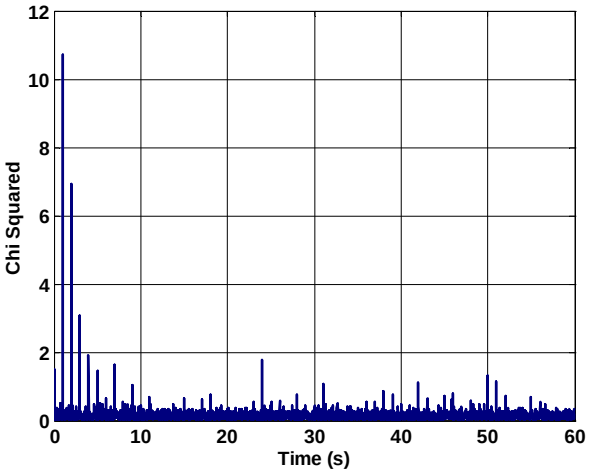

Figure 3-37: Chi-Squared Statistic for Underestimated Initial Position Error

Underestimated Measurement Noise

Simulations were conducted using underestimated noise, meaning that the filter was given a measurement noise covariance that was smaller than the actual measurement noise. Noise levels were maintained at their nominal levels while the filter parameters were reduced. The filter is reasonably robust in this respect.

For example, Figure 3-38 shows a plot of attitude error when all measurement noises are underestimated by a factor of 10. This results in more noise in the estimate. By underestimating the measurement noise, gains are two high, and the measurement noise is not successfully filtered. Convergence does occur, although the result is less accurate.

The excess measurement noise is quite obvious in the chi-squared statistic, shown in Figure 3-39. The large spikes correspond to the arrival of GPS measurements.

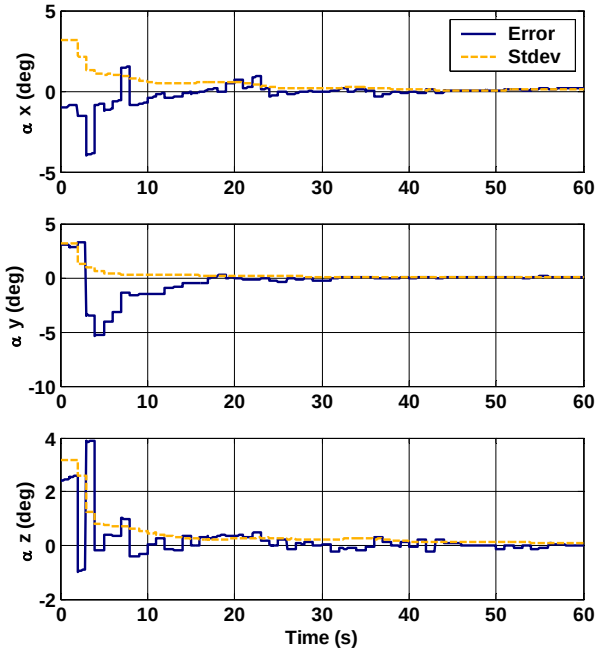

Figure 3-38: Attitude Accuracy with Underestimated Measurement Noise

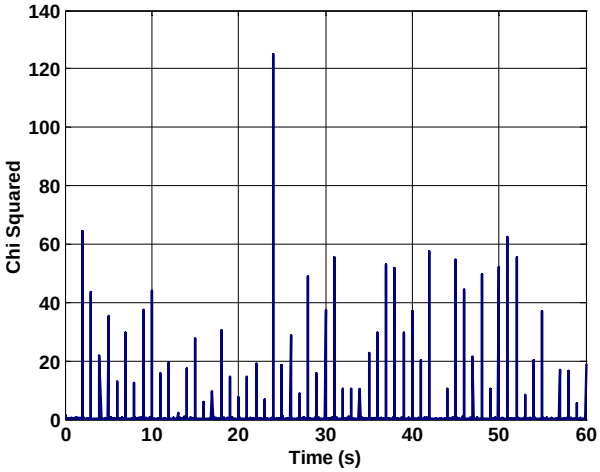

Figure 3-39: Chi-Squared Statistic for Underestimated Measurement Noise

Underestimated Process Noise

The filter is also reasonably robust to underestimated process noise. In these simulations, the filter process noise covariance is smaller than the actual process noise. The force and torque disturbances have the largest noise covariance and therefore have the greatest affect.

Most simulations show good results. Attitude accuracy is noticeably compromised at 100 times the process noise, as shown in Figure 3-40. This result is important, because it is

difficult to know the actual values for process noise, especially for force and torque disturbances.

The chi-squared statistic is plotted in Figure 3-41. The value is larger that the nominal simulation (Figure 3-32), but the average does not appear to grow over time.

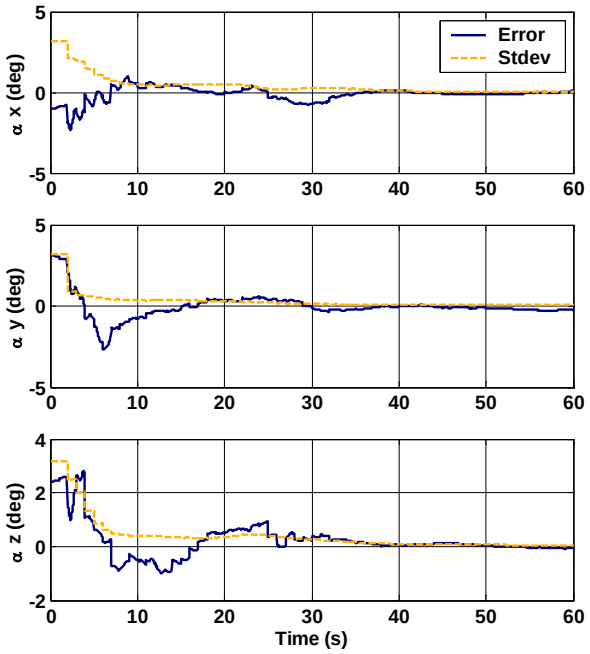

Figure 3-40: Attitude Response with Underestimated Process Noise (100x)

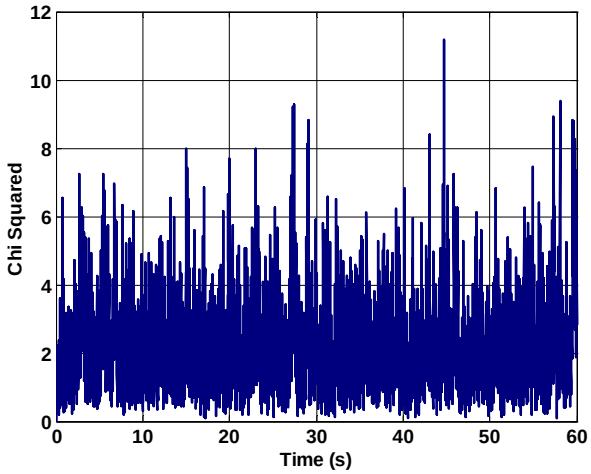

Figure 3-41: Chi-Squared Statistic with Underestimate Process Noise (100x)

Unmodeled Measurement Glitches

These simulations study the effect of a measurement glitch. This is an abrupt measurement error that is well beyond the noise model. Two types of glitches are studied: a point glitch, where there is only one bad measurement, and a step glitch, where a sensor unexplainably shifts its measurement value.

The effect of a single point GPS position glitch is shown in Figure 3-42. A measurement error of 10x was applied at $t = 10$. The result is a jump in the position estimate at that time, but recovery does occur. The attitude estimate, shown in Figure 3-43 is unaffected.

The chi-squared statistic is a excellent indicator of measurement glitches, as shown in Figure 3-44. Poor measurements of this type can be discarded on the basis of their chi-squared value.

The effect of a momentary gyroscope glitch is shown in Figure 3-45. Recovery occurs rather quickly.

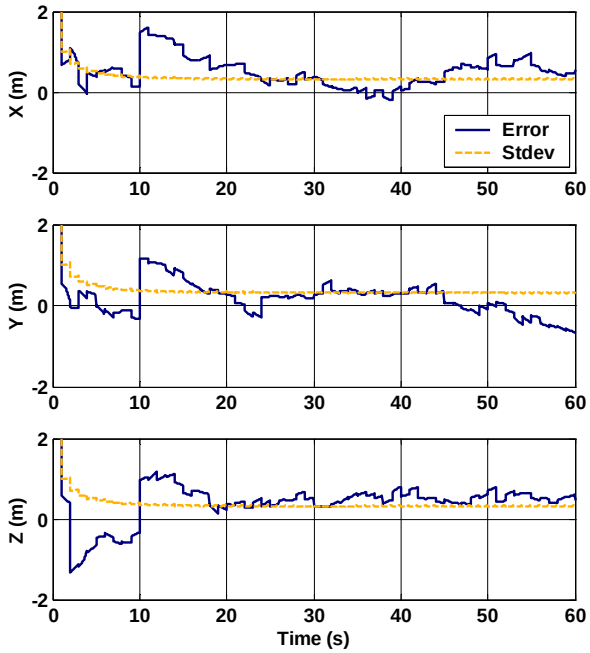

Figure 3-42: Position Estimate with GPS Position Glitch at t = 10s

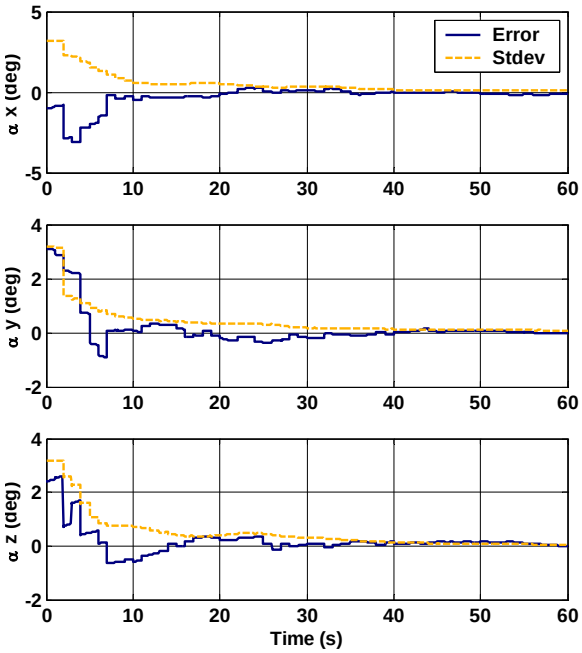

Figure 3-43: Attitude Estimate with GPS Position Glitch at t = 10s

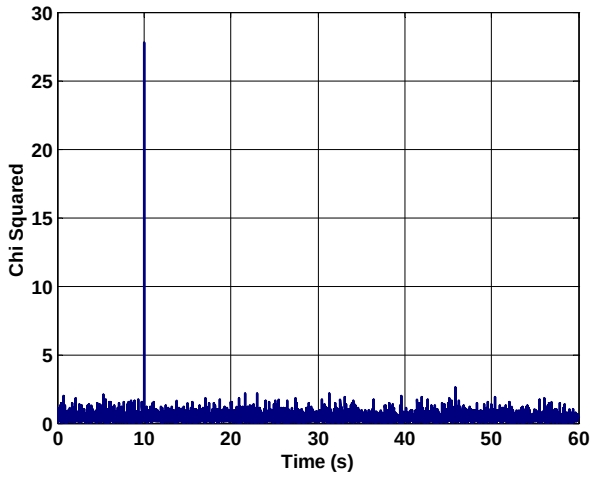

Figure 3-44: Chi-Squared Statistic with GPS Position Glitch at t = 10s

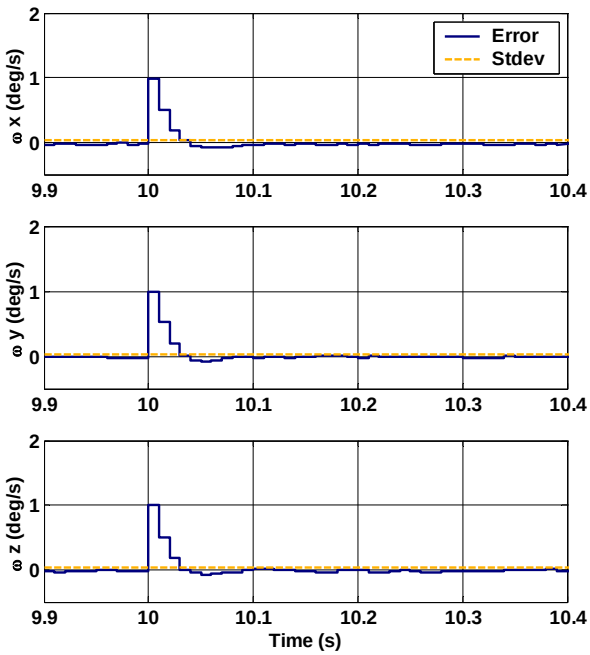

Figure 3-45: Body Rate Estimate during Gyroscope Glitch (100x)

A step glitch produces a different type of problem. Here, the measurement error persists, and the filter cannot ignore the new value. For example, a GPS solution may be altered when a new satellite comes into view and becomes part of the solution.

The effect of a persistent GPS Position error is shown below. The transition occurs at $t = 10$, and remains in error by 10 standard deviations. The position estimate, shown in Figure 3-47, gradually approaches the new value. This error has a small effect on the attitude estimate, shown in Figure 3-46. The chi-squared statistic, shown in Figure 3-48, gradually decreases as the filter converges on estimates that agree with the new measurements.

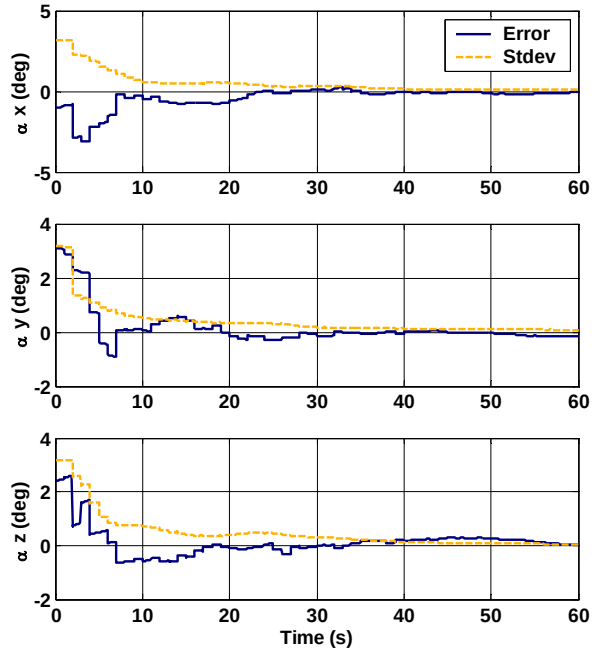

Figure 3-46: Attitude Error with GPS Position Step Glitch

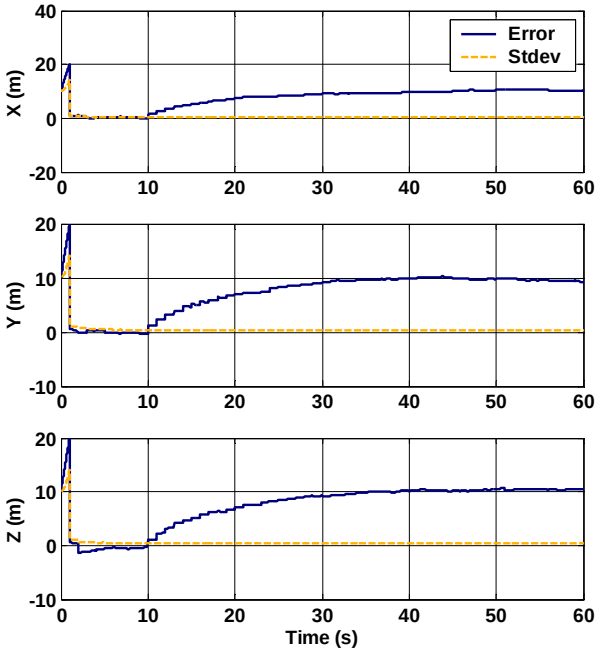

Figure 3-47: Position Error with GPS Position Step Glitch

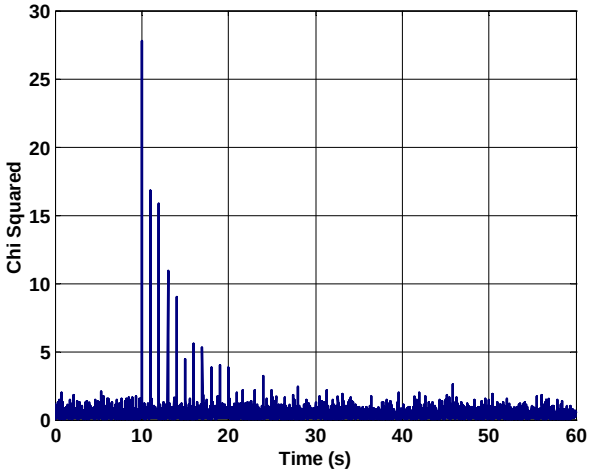

Figure 3-48: Chi-Squared Statistic with GPS Position Step Glitch

Data forgetting shows mixed results with these disturbances. For the most part, forgetting provides faster recovery because the glitch is more quickly ignored. However, this fast response also creates a greater effect at the time of the glitch. Since forgetting prefers recent data, this technique is beneficial when the glitch is old but harmful when the glitch is recent.

3.4.3 Additional Trajectories

The previous studies used a trajectory that was rich with information. This trajectory was somewhat contrived and not necessarily realistic. This section analyzes the effect of less rich trajectories with a particular interest in the estimate convergence.

Earth Stationary

As mentioned, the Earth stationary trajectory provides poor observability of attitude because the gravity vector can only be used to align in two dimensions. The remaining dimension must be aligned using Earth rate, which is very small compared to the gyroscope measurement accuracy.

The resulting attitude accuracy is shown in Figure 3-49. It is apparent that attitude does not converge in two dimensions. The steady error in α_x is the result of an unobserved accelerometer bias, and this behavior is unavoidable and entirely correct. However, the error in α_z persists while the estimated standard deviation converges. This situation is usually indicative of an error in filter construction or operation.

In this particular case, this error is caused by a fundamental problem of the Extended Kalman Filter. The convergence of the covariance is driven by the richness of the trajectory. More precisely, it is driven the richness of the estimated trajectory. For example, the observability of attitude was dependent on the time variation of $\hat{f}$. It is possible for noise in the state estimate to create this variation. This causes the filter covariance to converge too quickly, producing small Kalman gains, while the actual error has not converged at all.

This problem is only significant in situations of poor observability. A linearized Kalman filter, where the state estimate is not used as the point of linearization, has the advantage in this respect. This problem can be remedied approximately by using data forgetting or by adding artificial process noise to prevent the filter covariance from converging below the true value. The Earth stationary covariance analysis of Section 3.3.1 used a noise free trajectory, so the results represent performance without this problem. These results predicted attitude convergence in 10 to 20 minutes. The starting portions of the aircraft and booster trajectory in covariance analysis inadvertently contained some of this effect.

Earth stationary performance did show good estimation of gyroscope biases. Figure 3-50 shows that the bias estimate converges after 5 to 10 IMU measurements. The fixed-to-Earth measurement, at nominal accuracy, was used for these results.

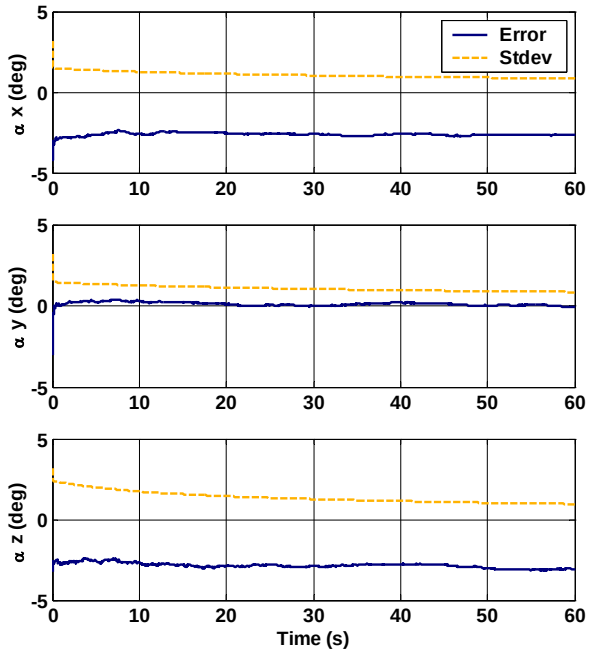

Figure 3-49: Attitude Error when Fixed to Ground

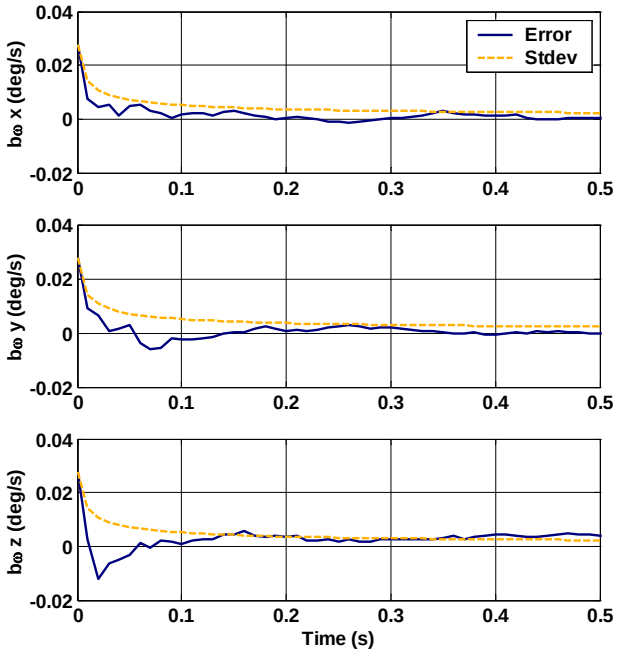

Figure 3-50: Bias Accuracy when Fixed-to-Earth

Modeled Aircraft Trajectory

The modeled aircraft trajectory is a simulated dataset that is designed to match the flight aircraft trajectory. It has been created using the rigid body dynamics model with force and torque values calculated from the flight measurements. This allows for a simulation with a known truth model and full control over noise parameters, while maintaining conditions that are very similar to the flight environment.

Bias values remain at 100x for this simulation. While this ultimately causes poorer accuracy, it shows that performance with large biases is successful in this flight environment. Predictions of filter accuracy in flight can be taken from the covariance analysis, which thoroughly explores variations in most parameters.

The attitude accuracy of this simulation is shown in Figure 3-51. Convergence is much better than the Earth stationary simulation, and comparable to the simulated rich trajectory. Note that attitude convergence is slow before takeoff (t = 1150). Alignment about the x and y body axes requires an estimation of the accelerometer biases. Alignment about the z axis before takeoff depends on Earth rate. When flight motion begins, the observability is sharply improved and convergence occurs more rapidly.

Note that z alignment continues to wander during flight in periods between maneuvers. This is in part because of the 100x gyroscope biases. However, this shows that simple flight paths will reduce accuracy.

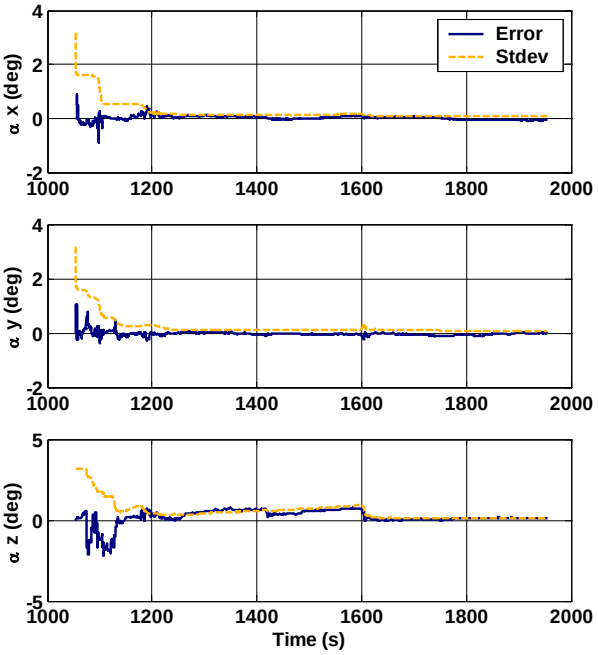

Figure 3-51: Attitude Accuracy using Modeled Aircraft Trajectory

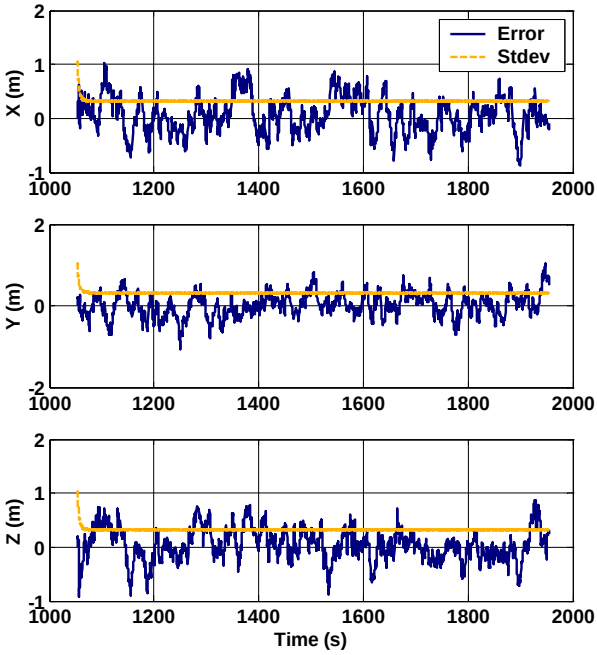

Figure 3-52: Position Accuracy using Modeled Aircraft Trajectory

Position accuracy is shown in Figure 3-52. While the estimate appears noisy on this time scale, the error values are within the standard deviation most of the time. This suggests statistical agreement between the error and filter covariance.

Modeled Booster Trajectory

The modeled booster trajectory is a creation similar to the modeled aircraft trajectory, but is based on the flight booster data. Once again, the rigid body dynamics model is used to generate a simulation with known truth and adjustable noise parameters. Biases remain at 100x.

The attitude performance is shown in Figure 3-53, and shows good convergence after liftoff ($t = 0$). Force is predominately in the axial direction before launch and during flight, so alignment about the x axis is the most difficult. Position accuracy is shown in Figure 3-54 and is in agreement with the estimated standard deviation.

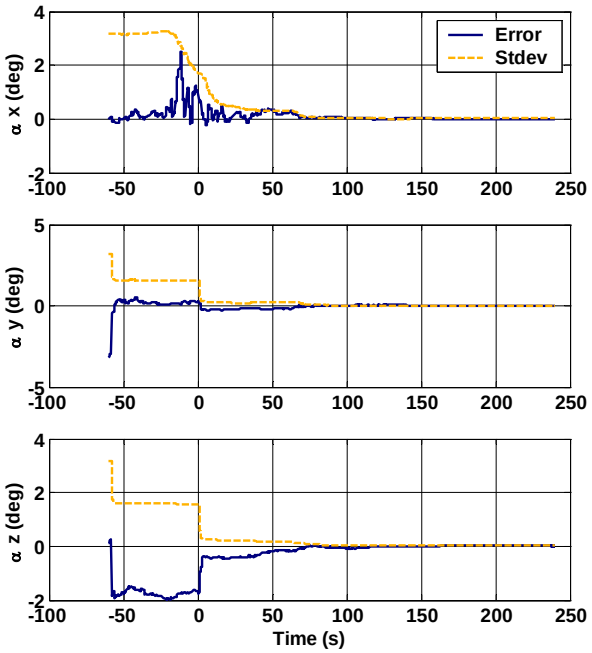

Figure 3-53: Attitude Accuracy using Modeled Booster Trajectory

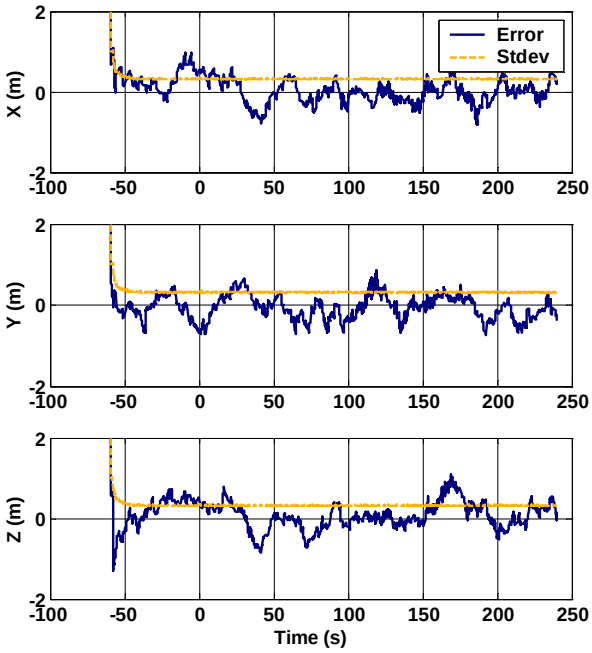

Figure 3-54: Position Accuracy Using Modeled Booster Trajectory

3.4.4 Omitting Bias States

It is useful to study the effect of removing the bias states from the model. In the following simulations, biases are still present in the simulated gyroscope and accelerometer measurements and in the gravity model. However, the Kalman filter does not have bias states to represent these quantities. As a result, the noise model is incorrect.

Two levels of biases are studied. First, the previous nominal values are used. These values are 100 times the actual hardware values, and have a significant effect on filter performance. Using the larger biases is instructive for studying the effect of unmodeled biases. Afterwards, the simulations are rerun using the actual values, which have substantially less impact.

A distinctive characteristic of these simulations is shown in Figure 3-55. While the estimated standard deviation converges, the estimated state has divergent tendencies. In this case, the accelerometer measurements and gravity model are biased, so the position estimate diverges between GPS measurements. The GPS corrections attempt to restore the position estimate, but the gain is too small.

The result is much worse for attitude estimation, as shown in Figure 3-56. Drift here comes predominantly from the gyroscope bias. Since attitude suffers from poor observability, there is little to correct the attitude error.

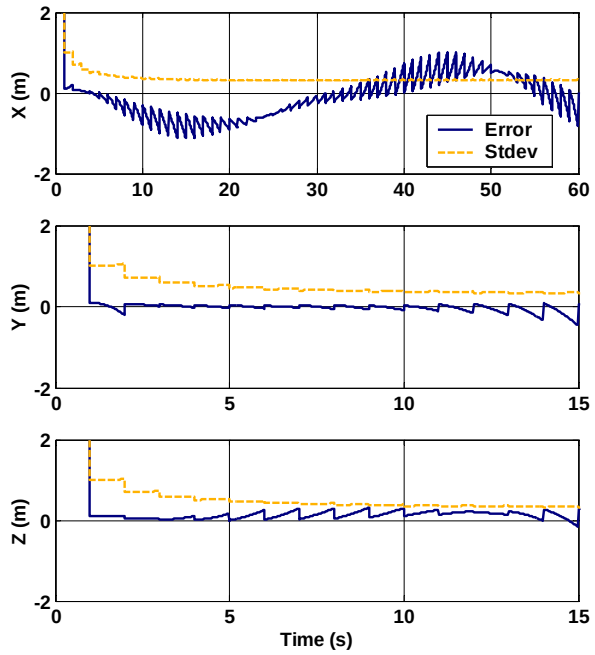

Figure 3-55: Position Response with Unmodeled Biases

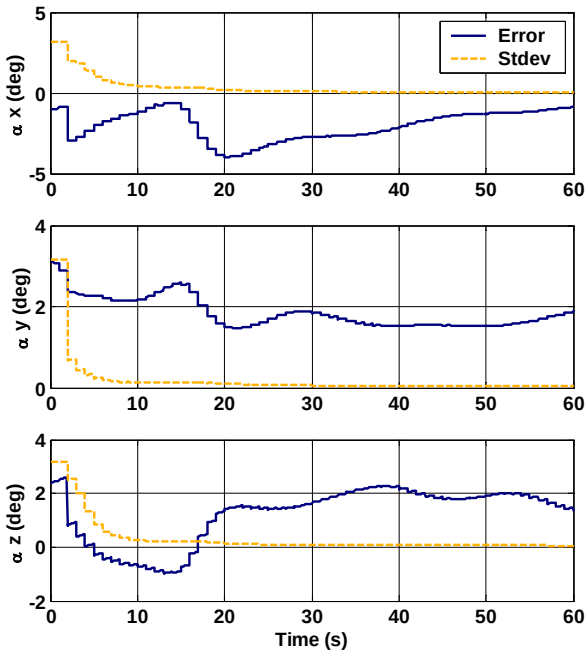

Figure 3-56: Attitude Response with Unmodeled Biases

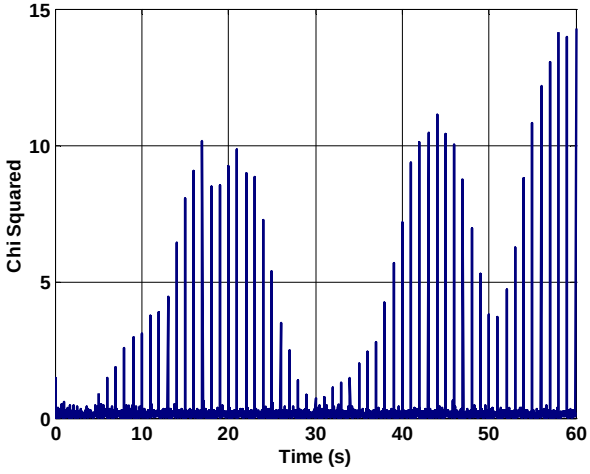

Figure 3-57: Chi-Squared Statistic with Large (100x) Unmodeled Biases

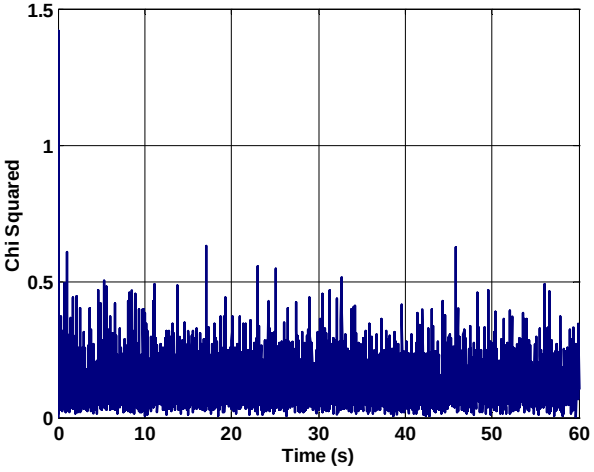

Figure 3-58: Chi-Squared Statistic for Small (1x) Unmodeled Biases

The Chi-Squared statistic is a good tool for diagnosing this type of problem. As shown in Figure 3-57, the measurements do not agree with the model. The covariance converges but the biases persist, leading to increasingly inconsistent measurements.

Simulations also show that data forgetting can not solve this type of modeling error. This is because the modeling error has an affect on data for all time. The plots are very similar and show a slight increase in the estimated standard deviation. This causes increased gains, which can actually worsen the problem.

Using the actual bias levels, one hundredth the nominal values, there is little performance degradation. The chi-squared statistic, shown in Figure 3-58, suggests that the biased measurements still fit this simplified model. The state error response is indistinguishable from the response of the full model.

It should be concluded that unmodeled states may or may not be significant. Since states variables bear a large computational penalty, it may be wise to use the smallest model that produces a good chi-squared plot.

An alternative technique is to compensate the biases independently from the Kalman filter. This may be successful is reducing the effective bias to values that can be ignored by the Kalman filter.

3.4.5 Measurement Dropouts

Another test of filter robustness is the response to measurement dropouts. These simulations explore dropouts in the IMU, GPS, and in both simultaneously. For each, there is a time window during which the measurements are not received. The filter continues to update the state and to process any other measurements that are available.

For these studies, the modeled aircraft trajectory is used, because it provides a realistic flight environment with a known truth model. The measurement dropout begins at time $t = 1200$, which is about 50 seconds after takeoff when the aircraft is maneuvering moderately. Dropouts last from 10 to 60 seconds. Biases remain at 100x for these simulations.

The response to a 60 second GPS dropout is studied first. The attitude accuracy is plotted in Figure 3-59 and the position accuracy is plotted in Figure 3-60. The position error and its estimated standard deviation both increase abruptly during this time period because the accelerometers are the only source of position information. In effect, the Kalman Filter captures the drift of the inertial solution. This is highly beneficial, because it leads to a fast recovery when the GPS is restored.

The attitude error is largely unaffected. The attitude error and standard deviation are increased because there are no GPS corrections for the attitude. This also makes the attitude estimate more smooth. The decrease in the standard deviation of α_y is the result of vehicle rotation.

The effect of a 60 second IMU dropout is more severe. The attitude is plotted in Figure 3-61. The attitude performance is very poor during the dropout, showing errors of 30° to 50°. The standard deviation is seen to rise greatly because attitude is very difficult to observe without gyroscopes. When the IMU measurement is restored, performance returns to normal. This is quite remarkable considering the small angle approximation.

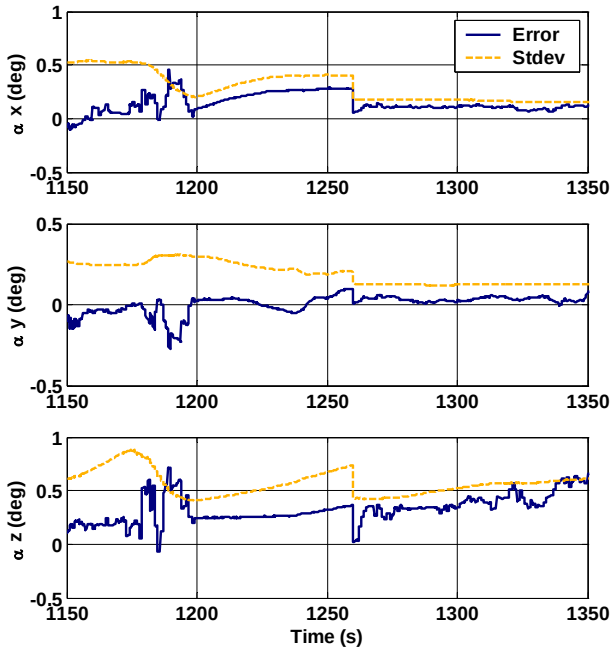

Figure 3-59: Attitude Accuracy during GPS Dropout

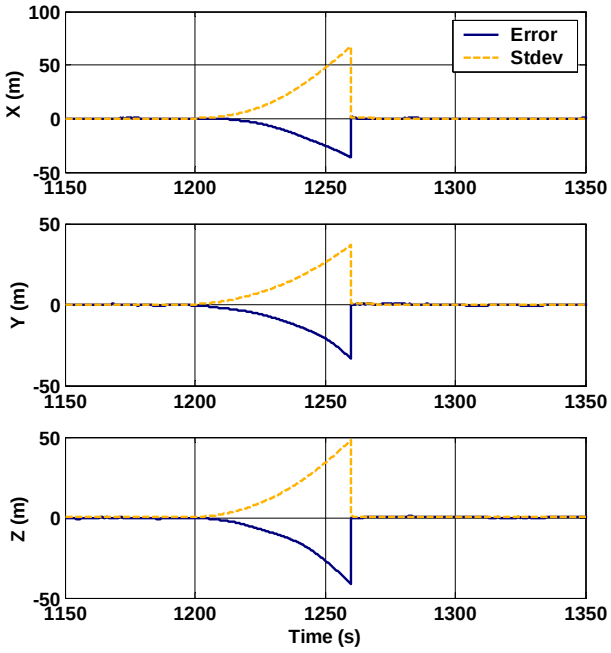

Figure 3-60: Position Accuracy during GPS Dropout

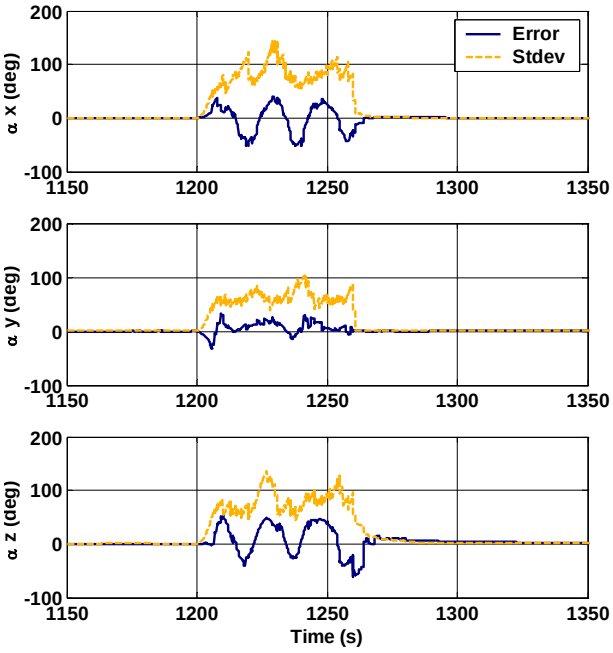

Figure 3-61: Attitude Accuracy during IMU Dropout

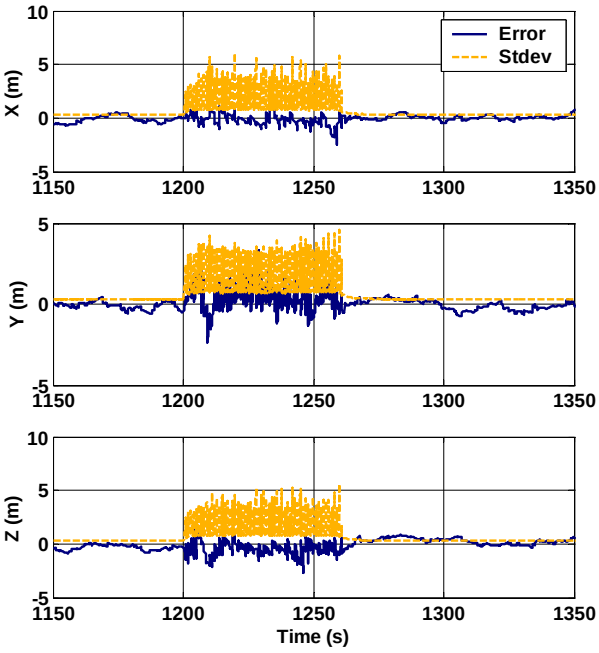

Figure 3-62: Position Accuracy during IMU Dropout

The position accuracy for the IMU dropout is plotted in Figure 3-62. The position estimate is maintained adequately throughout the dropout. The position uncertainty now grows rapidly

between GPS measurements, causing the messy plot, because the acceleration is not measured. After the dropout, behavior returns to normal.

Finally, the effect of a simultaneous GPS and IMU dropout is studied. The length of the dropout has been reduced to 10 seconds because the filter can not handle the longer dropout. The attitude is shown Figure 3-63. The estimated standard deviation grows rapidly and the error drifts where it may. However, a full recovery is achieved once again when the measurements return.

The position accuracy is plotted in Figure 3-64. This shows a similar increase in error and standard deviation. With no measurements of position, velocity, or acceleration, the filter can only use its knowledge of the force dynamics. In this model, force decays towards zero when unobserved. It may have been beneficial to instead have force approaching the reaction to gravity.

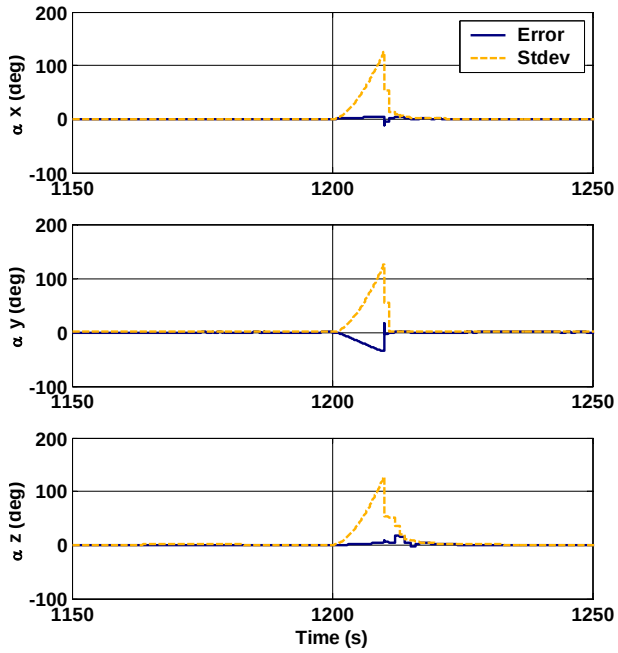

Figure 3-63: Attitude Accuracy during IMU and GPS dropout

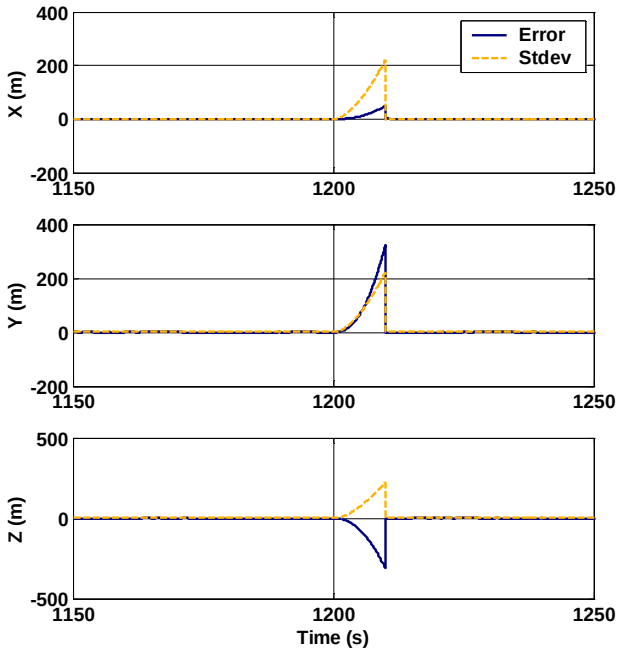

Figure 3-64: Position Accuracy during IMU and GPS dropout

These simulations showed reasonable performance in unreasonable circumstances. Performance could probably be improved by adjusting noise parameters and decay rates. In the absence of measurement information, the system model accuracy is more critical.

It is tempting to suggest that the Kalman filter should be shut down during measurement dropouts. For example, the navigation system could revert to an inertial solution during a GPS dropout. However, the Kalman filter still provides the optimal estimate during the dropout, and can provide several benefits. Returning to the plots in Figure 3-59 and Figure 3-60, the Kalman filter maintains the attitude error through the dropout. The position drifted 300 meters in 60 seconds, but this is small considering that the IMU was biased by 0.29 m/s^2.

Finally, the Kalman filter calculates the proper increase in covariance for when the measurement is restored.

Despite these benefits, it is only a matter of time before the covariance grows to be too large for numerical computation. This can corrupt the estimates of observable states and ultimately lead to a numerical failure. This can be addressed using several techniques.

To the extent that the increase in covariance is real, the uncertainty in the estimate is becoming too large for the linearization of the Extended Kalman Filter to be valid. In this case, the computation must be handed over to another algorithm with nonlinear capability. This could be the same algorithm that was used in the initialization routine. When this routine finishes, possibly after the dropout, the EKF should be restarted with an initial state covariance representative of the nonlinear algorithm's accuracy. In this architecture, the EKF is protected by a nonlinear algorithm so that the linearization is always valid and the covariance remains bounded.

It is also possible that the increase in covariance is the result of fictitious noise or data forgetting, which may have been employed originally to make the filter more responsive. During the dropout, all data should be remembered, and noise parameters should be returned to their actual values.

Beyond this, it is possible to manipulate the Kalman Filter more intrusively. Unobservable state variables can be frozen by zeroing their process noise and the corresponding rows and columns of the covariance matrix. Limits on the covariance can be imposed, but care should be taken that the covariance matrix remains symmetric positive definite. While these techniques can provide benefits, they are ad hoc and are conceptually flawed. If the model is correct and the covariance is large, then there is no accurate solution. Furthermore, the most accurate solution, in terms of minimum error variance, is given by the unaltered Kalman filter.

3.5 Flight Data Simulation

Real flight data is available from several programs at MIT Lincoln Laboratory. Simulation using this data provides a test of filter performance in a real flight environment. Although simulated data is more flexible, flight data contains the actual sensor noise and disturbances.

Datasets have been recorded from sensors aboard an aircraft and a booster rocket. This data is then loaded from file and replayed for simulation. This provides a substitute for flying the actual filter.

The difficulty with this data is that the actual state is not known exactly. Both datasets contain a reference solution from another navigation system. However, the accuracy of these solutions could not be precisely established and errors in these solutions were visible.

Without a reliable truth solution, assessment of accuracy becomes a more subjective matter. The inertial solution can be used as a short-term reference, but will have noticeable drift by the end of flight. Testing that the filter remains healthy and converges is also an important part of this simulation.

In general, the modeled data simulations provides a better estimation of filter accuracy. The simulations here show that the filter functions reliably in the flight environment. They also show the validity of various modeling assumptions used in the previous simulations.

3.5.1 Aircraft

The aircraft dataset was measured aboard a small airplane during a short flight. The data includes takeoff and landing as well as some airborne maneuvers and ground taxiing. Data was recorded from a Litton LN200 IMU and a NavCom SF-2050M GPS receiver. Both the raw GPS measurements and the position and velocity solutions were available. For the purpose of computing a Differential GPS solution, a second set of GPS data was recorded from a stationary GPS receiver located at a nearby ground station. For this simulation, only the GPS position and velocity and the IMU measurements are used by the filter.

This dataset also contains a reference solution. This solution has been generated using a software package called GrafNav, by Waypoint Consulting, that post-processes the measurements. This package uses a forward-backward smoothing algorithm and makes full use of both GPS receivers. Unfortunately, this solution proved to be unreliable. As will be analyzed shortly, there were several obvious glitches in the solution. As a result, this “high-quality” solution can only be used as an approximate reference.

The filter estimates of attitude and position are shown in Figure 3-65 and Figure 3-66, respectively. Attitude has been displayed in NED Euler angles and position has been converted to geodetic coordinates. These results are in reasonable agreement with the reference and inertial solution. The flight maneuvers are readily apparent in this data. Takeoff and landing occur at $t = 1150$ and $t = 2800$, respectively. Several large turns in yaw are also present.

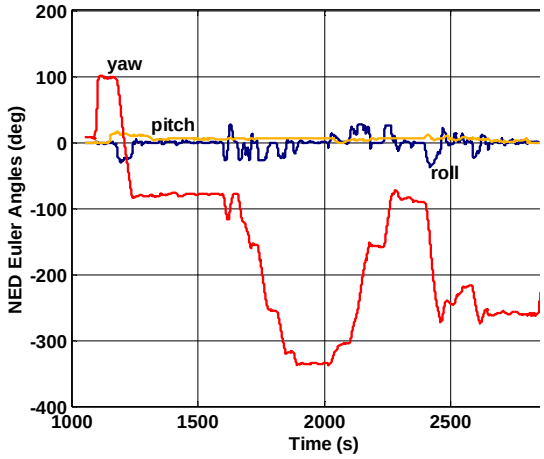

Figure 3-65: Aircraft Attitude

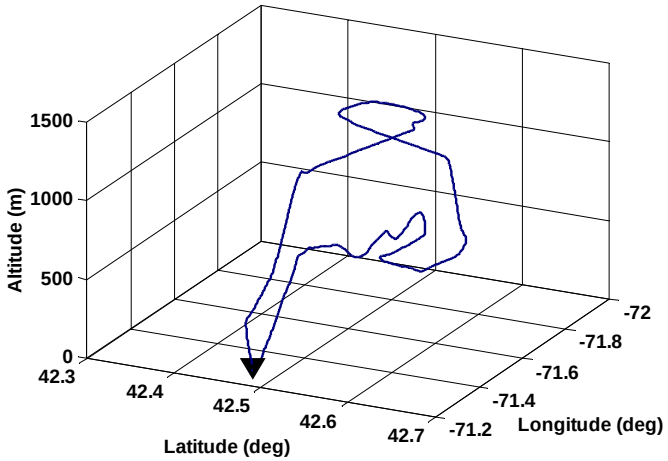

Figure 3-66: Aircraft Position

The filter estimates of torque and force disturbances are shown in Figure 3-67. Disturbances are largest during takeoff and landing. The *z* component of the force bias opposes gravity throughout the flight. Additional forces are seen during maneuvers.

Biases have also been estimated and are plotted in Figure 3-68. Although there is no way of knowing the actual bias values, these estimates seem reasonable. There is some divergence in the accelerometer and gravity model biases, but the bias decay rates will eventually keep these estimates bounded. These bias estimates may have been influences by the presence of other types of errors, such as IMU scaling errors or GPS delays.

The attitude error is plotted in Figure 3-69. This error is often larger than the estimated standard deviation, especially in α_y and near the end of simulation. This error is the difference between my filter estimate and the reference solution. As a result, errors can be caused by either solution. There is reason to believe that much of this error was created by errors in the reference solution. The abrupt spikes in the error are present in the reference solution. Also, the reference solution contained other noticeable glitches which will by analyzed shortly.

The position error is shown in Figure 3-70. This error is in good proportion to the estimated standard deviation.

The chi-squared statistic is shown in Figure 3-71. The value remains small, but shows groups of irregular measurements. Many of these irregularities occur during maneuvers, when the aircraft forces and torques are the largest.

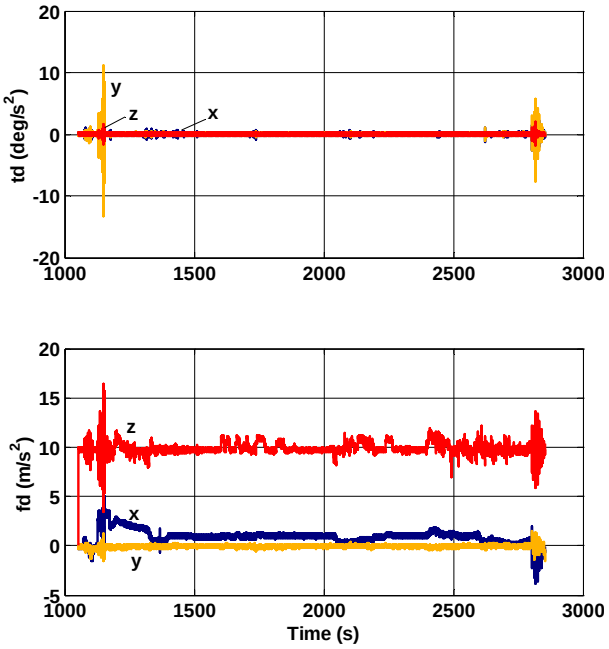

Figure 3-67: Estimate of Torque and Force Disturbances

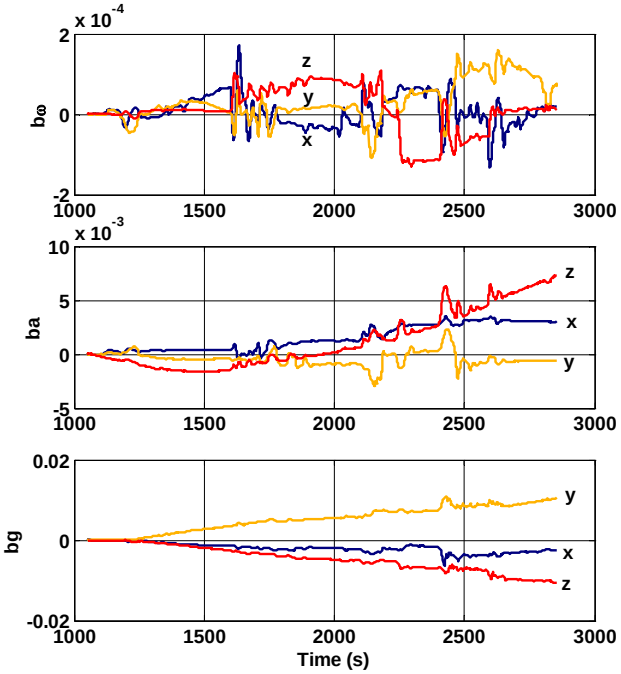

Figure 3-68: Bias Estimate in Gyroscopes, Accelerometers, and Gravity Model

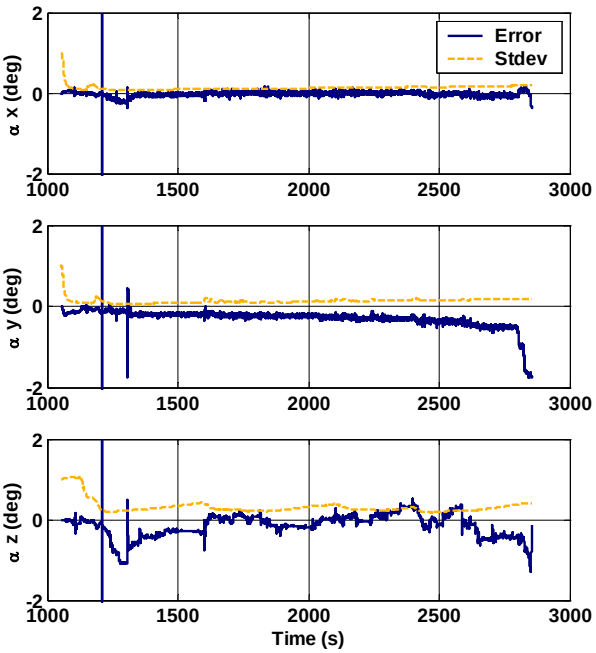

Figure 3-69: Attitude Estimate Error

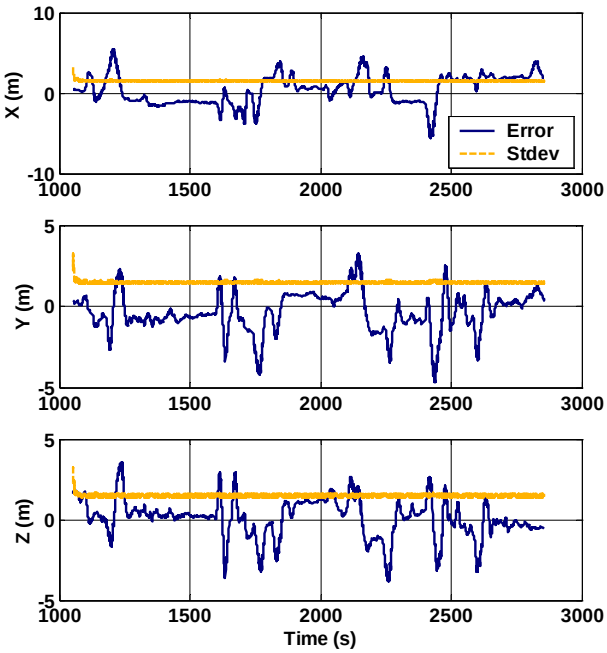

Figure 3-70: Position Estimate Error

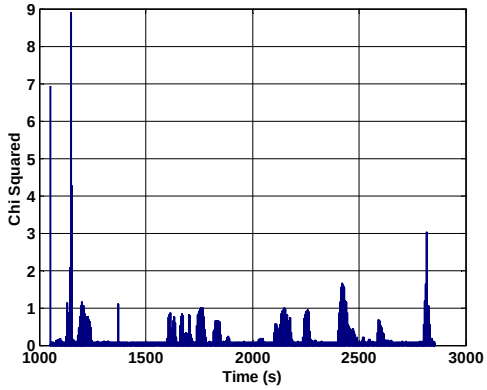

Figure 3-71: Chi-Squared Statistic

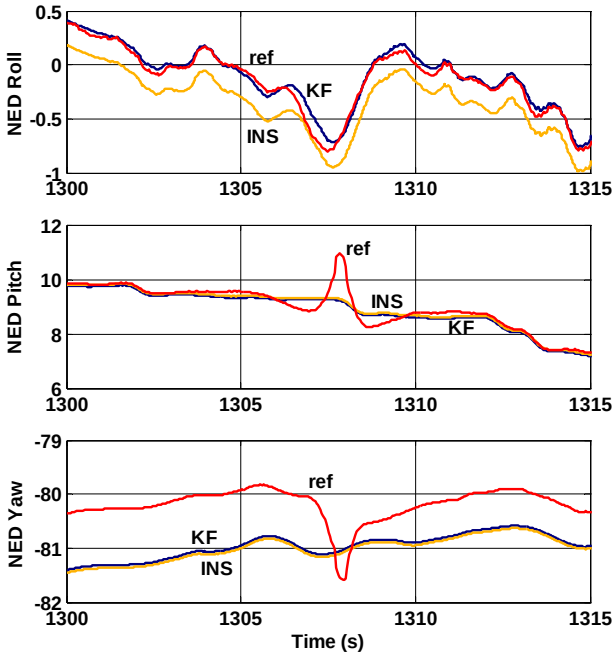

Figure 3-72: Glitch in Reference Solution

Figure 3-72 shows what is undeniably a glitch in the reference solution. On this plot are the estimates of the Kalman filter (KF), the inertial solution (INS), and the reference solution (ref). The Kalman filter and inertial solutions follow each other precisely, despite an offset in roll. The reference solution, however, deviates greatly in the time 1307 to 1308. The reference solution does not recover from this glitch, leaving almost 1º of unexplained yaw rotation.

This LN200 IMU has a drift of 1º per hour, specified at 1 standard deviation, so it is statistically impossible that the INS solution had this much error. In addition, direct analysis of the measured body rates did not show this motion. It is therefore concluded that this reference solution is not accurate. The causes of this error in the provided reference solution are outside the scope of this thesis and have not been investigated further.

3.5.2 Booster Rocket

The booster rocket flight data provides measurements from the harsh environment of a launch vehicle. As mentioned previously, this data contains a period before launch, during boost phase, and after boost phase. These present drastically different operating conditions.

This dataset includes only the raw measurements from the IMU, which was a Litton LN200. It contains a reference solution that was computed by an onboard Litton LN100, which is a full GPS-aided navigation system. The GPS position and velocity solutions were not available, so these measurements were simulated. These simulated measurements were generated by adding noise to the position and velocity of the reference solution.

The quality of the reference solution is reasonably good, but there is difficulty aligning the solution times to the measurement times. While both the IMU measurements and the reference solution contain time markers, there is evidently some confusion in the time offset between these two data sources. For this reason, the reference data source is once again approximate.

The estimated attitude is shown in Figure 3-73 and the estimated position is shown in Figure 3-74. These show the gradual arc of the booster trajectory. The booster force profile, which contained two stages, is evident in the estimate of force disturbance, shown in Figure 3-75. Following the booster phase, force drops to near zero. Returning to Figure 3-73, several attitude maneuvers are then performed.

Bias estimates are shown in Figure 3-76. It is not clear that biases actual converge, but they do seem to remain bounded. This may be because there is primarily axial acceleration. Also, the great variation in force magnitude would augment the effect of scaling errors, which were omitted from the filter model.

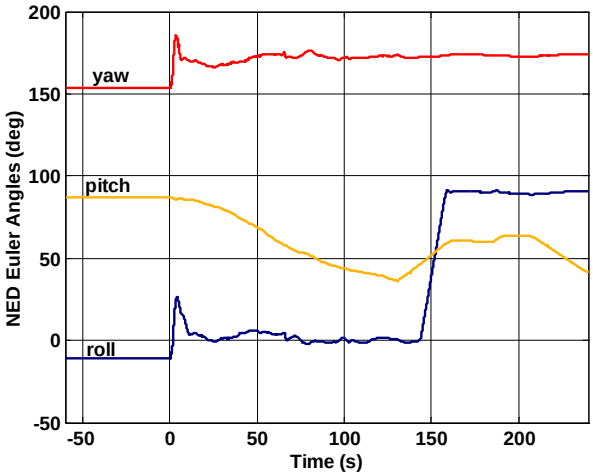

Figure 3-73: Booster Attitude

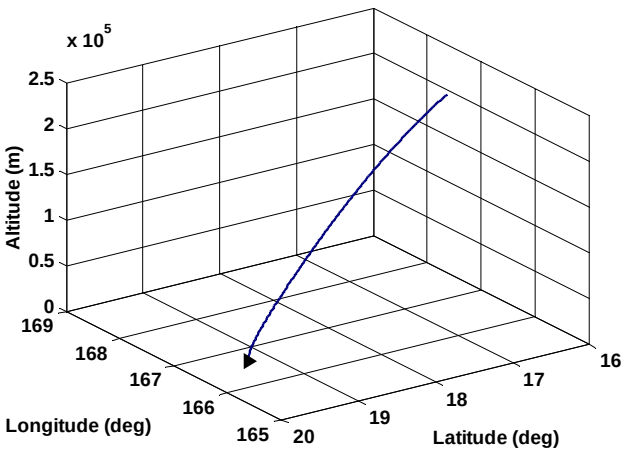

Figure 3-74: Booster Position

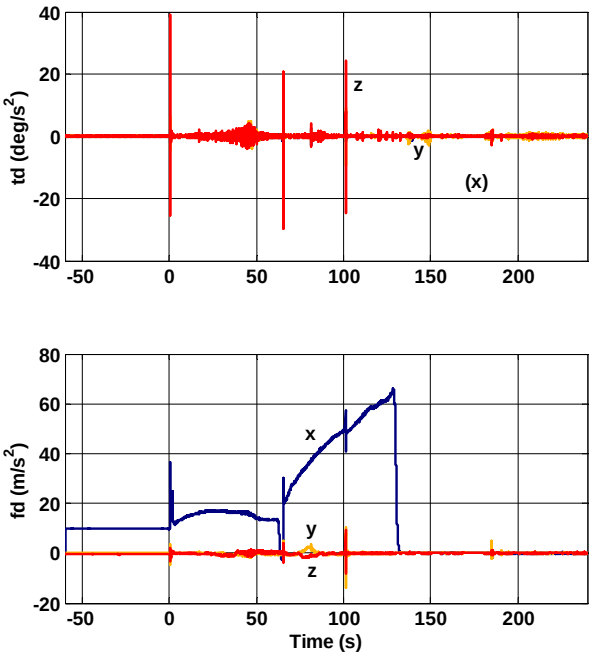

Figure 3-75: Estimated Force and Torque Disturbance

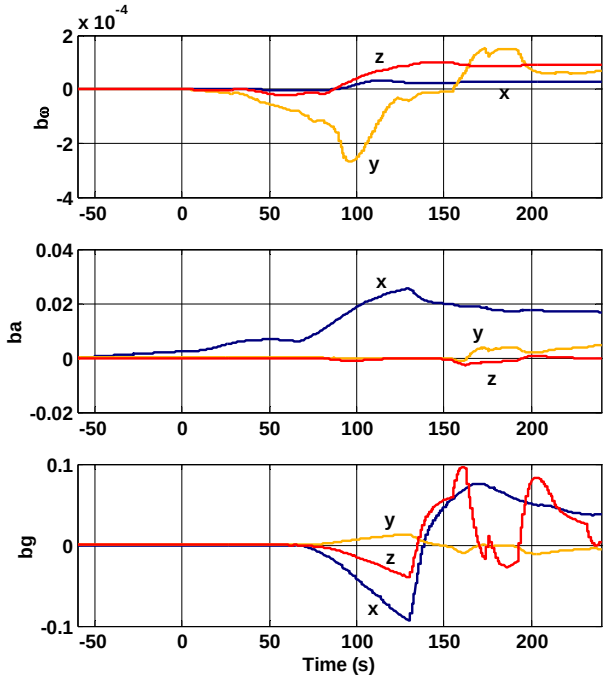

Figure 3-76: Bias Estimates

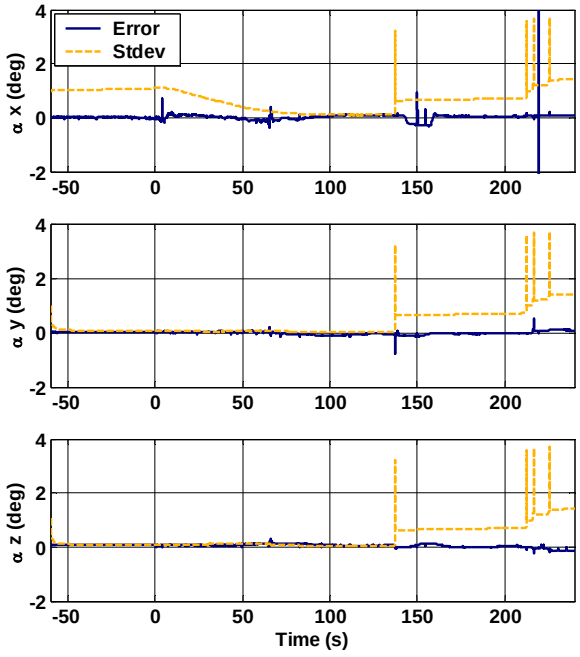

Figure 3-77: Attitude Estimation Error

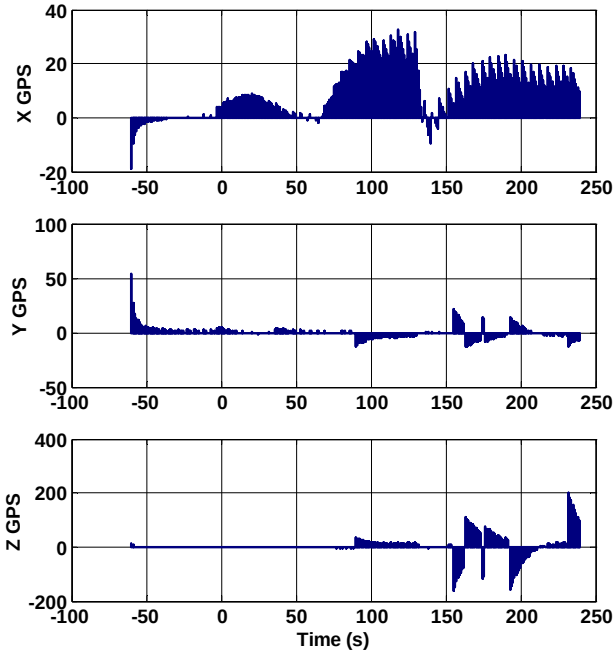

Figure 3-78: GPS Position Error

The attitude estimation error is plotted in Figure 3-77. Several spikes are present in the estimated attitude standard deviation, and sometimes in the attitude error itself. These

correspond to dropouts in the dataset. The actual IMU measurements did not dropout, but the telemetry used to transmit this data back to Earth experienced brief losses in transmission. As a result, this filter is confronted with short IMU dropouts.

The position error is plotted in Figure 3-78, in the form of the GPS measurement error. This error is significantly worse than the GPS accuracy, but this can be attributed to several factors. Recall that the GPS measurement that is used in this simulation is actually manufactured from the reference solution. The reference solution is imprecisely time alignment, and is calculated at discrete time intervals. During periods of high velocity, this inaccuracy in time causes large inaccuracies in position. During periods of high acceleration, there may have been additional lags in the onboard GPS solution.

The chi-squared statistic is shown in Figure 3-79. The spikes correspond to drastic changes in the force, such as liftoff. Ideally, these spikes can be reduced by improving the system model. However, it is difficult to accommodate the great variation in noise characteristics that this trajectory presents. Experiments have been conducted using scheduling and adaptive estimation of the noise characteristics. While these techniques are generally successful, more development effort is required.

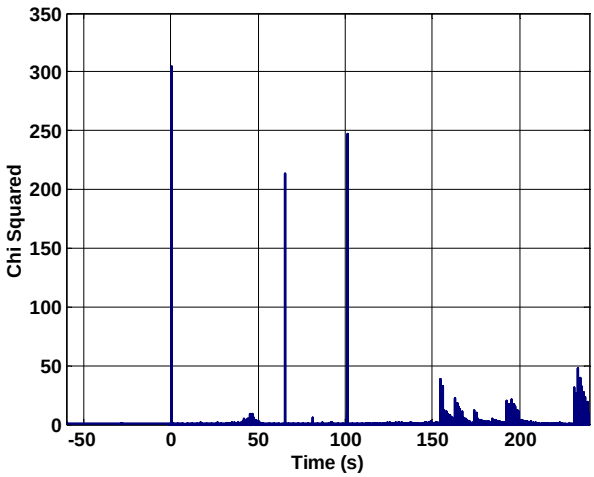

Figure 3-79: Chi-Squared Statistic

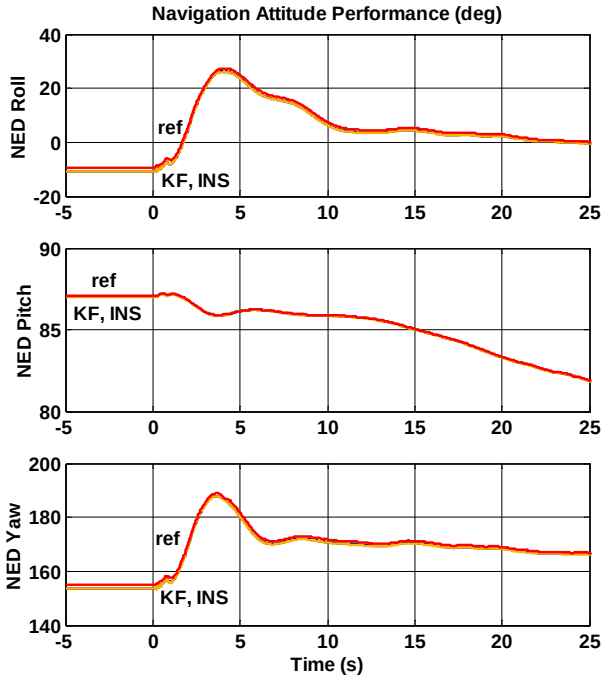

Figure 3-80: Solution Comparison during Launch

3.6 Initialization

The initializer provides nonlinear convergence to bring a large initial error into the capture range of the Kalman Filter. This is accomplished by processing the first group of

measurements using a different algorithm. This algorithm is less accurate than the Kalman filter, but has guaranteed stability for large errors.

For most states, this initialization is trivial and can be performed by averaging the first few measurements with the proper coordinate conversions. For example, the initial position can be calculated from the GPS measurements. Attitude, however, requires a special alignment routine. When the vehicle is fixed-to-Earth, the gyrocompassing routine is employed.

The performance of the gyrocompassing routine has been simulated using the fixed-to-Earth model, and is shown in Figure 3-81. This implementation uses a 1 second time constant for gravity vector alignment, and a 30 second time constant for Earth rate alignment. The body rate measurements are filtered using a discrete filter with an equivalent time constant of 10 seconds. The Earth rate alignment is delayed by 30 seconds so that this filter converges before the earth rate corrections are applied.

It can be seen that the gravity alignment procedure is very fast and accurate. The Earth rate alignment requires a longer time, and is greatly influenced by the gyroscope noise. In tuning the time constants, there is a tradeoff between accuracy and response time. Simulation shows a subsequent successful handoff to the Kalman filter.

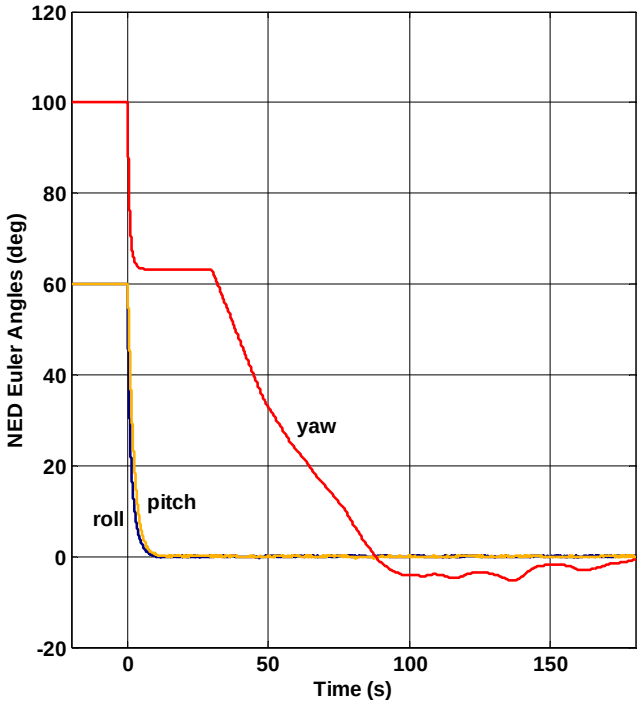

Figure 3-81: Initialization using Gyrocompassing

This completes the simulations analysis. We now proceed to study the hardware implementation.

4. Hardware

A hardware prototype has been created in order to test this filter in a real-time environment on embedded hardware. This explores the feasibility of this algorithm in the presence of real computational constraints. Ultimately, it has exposed several practical issues that are important in the design of this system.

Section 4.1 begins with a description of the hardware components. This includes the processor, IMU, and GPS receiver. Pictures of the major components are presented.

Section 4.2 describes the general process for implementing the software on the hardware. This begins by describing a technique for automatic code generation, so that the simulation models are ported into C code. However, this technique produces inefficient code and methods of improving the computational efficiency are required. These efforts and the resulting algorithms are detailed in Section 4.3.

Section 4.4 presents the results of hardware testing. Full flight testing of the hardware was not included in this project, due to resource and time constraints. Instead, the previous flight data simulations serve to validate this design in the flight environment. Hardware testing is designed to establish that the hardware performs similarly to the simulation.

As a result, hardware testing is limited to benchtop testing. The GPS antenna is fixed in position, while the IMU can be moved and rotated. This provides full testing of the Earth stationary trajectory and moderate testing of attitude performance. Further testing in real flight environments is recommended.

4.1 Description of Hardware Components

This hardware implementation uses high performance components for computation and measurement. These components have been chosen because of their history and future use on programs at MIT Lincoln Laboratory. The use of familiar components greatly accelerated the process of implementation. These components have been available as spares and development units from past programs.

The central processing is performed by a Texas Instruments SMJ320C6701 Digital Signal Processor (DSP). This processor contains dedicated hardware for floating point arithmetic, which has been configured for 32-bit computation. It also features a Very Long Instruction Word architecture with multiple data paths and functional units, so that up to 8 instructions can be performed in parallel during a single clock cycle. This component features a variety of on-chip peripherals including timers, serial ports, program and data memory, and an external memory interface. This particular processor operates at a speed of 132 MHz and provides 64 KB program cache and 64 KB internal data memory.

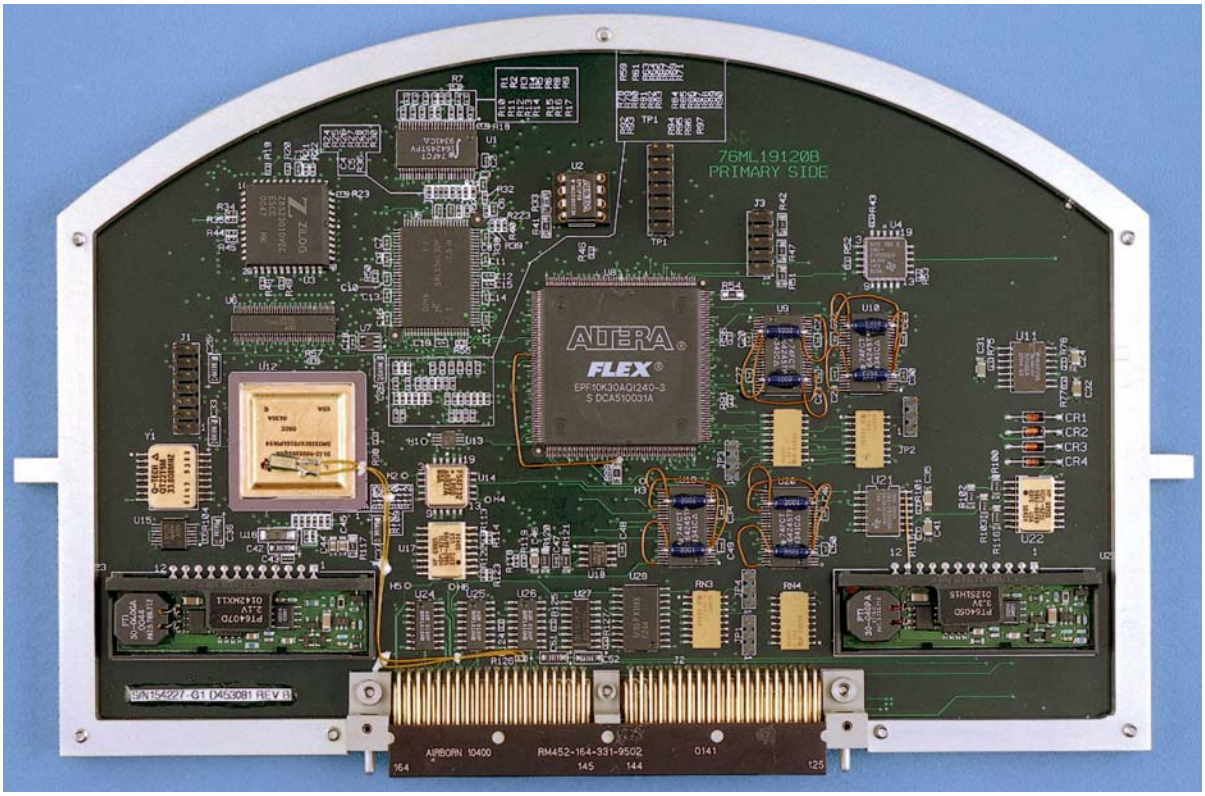

Figure 4-1: Processor Board

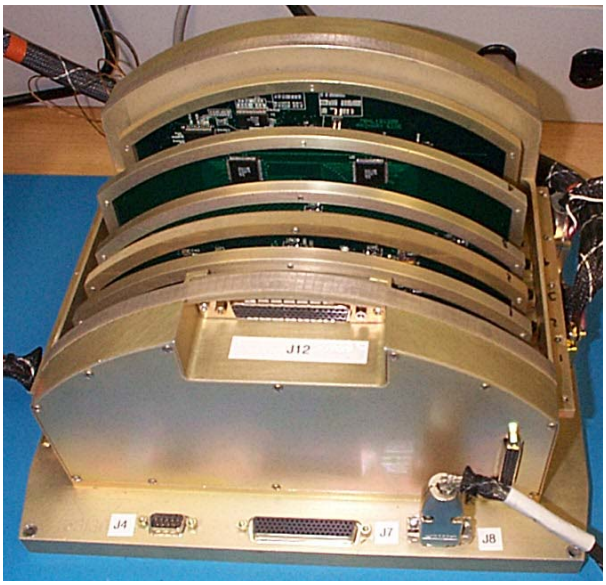

Figure 4-2: Card Cage for Processor Board and Power Electronics

This DSP is featured on a processor board that was developed at MIT Lincoln Laboratory by Deborah Blanchard for high performance embedded control and imaging applications. Figure 4-1 shows a picture of the primary side of this circuit board. The processor board contains components for digital IO, a 2 channel UART, and additional memory. The memory includes 2 MB FLASH for nonvolatile storage, 8 MB Dynamic RAM, and 1 MB

Static RAM. Peripheral addressing and certain other functions are performed by an ALTERA Field Programmable Gate Array (FPGA). The processor board is housed in a card cage, shown in Figure 4-2, which contains additional electronics for power distribution and data recording.

The IMU for this project is a Litton LN200, shown in Figure 4-3, featuring three fiber optic gyroscopes and three proof mass accelerometers. This unit is a high-performance, navigation-quality IMU. Specifications were provided with the nominal simulations parameters in Table 3-5. Data messages are sent to the DSP using a high speed serial data link that is processed by the FPGA.

The GPS receiver is an Ashtech G10 HDMA (High Dynamics and Missile Applications) receiver, shown in Figure 4-4. This unit makes full use of the military GPS frequency and encoding. It provides output of the position and velocity solution at 10 Hz in a serial data message format.

Figure 4-3: LN200 IMU

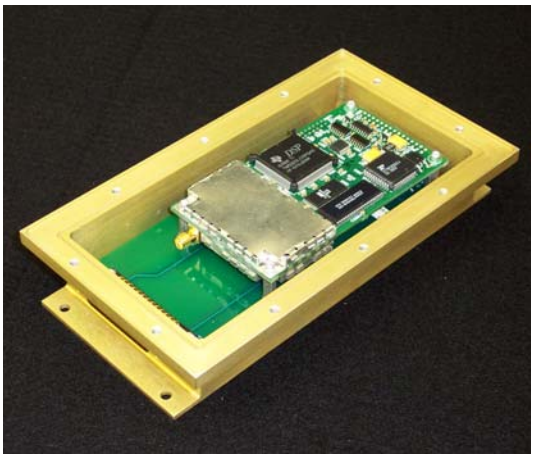

Figure 4-4: Ashtech G10 HDMA GPS Receiver and Chassis

4.2 Software Implementation

Implementation on the hardware requires porting the algorithm to DSP-specific instructions. This is done using the process shown in Figure 4-5. The Simulink model is first converted to C code and then the C code is compiled to produce the executable for the DSP.

The conversion from Simulink to C code is done automatically using a product by The MathWorks called Real-Time Workshop (RTW). With this method, there is no potential for error in recoding the algorithm. Effectively, the same filter block that is tested in simulation is converted directly to C code.

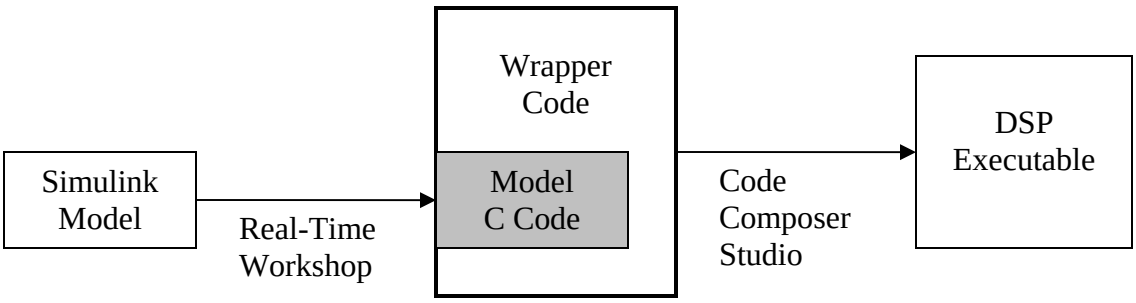

Figure 4-5: Software Implementation Process

Real-Time Workshop creates two top-level C functions for initializing and updating the model. The update function is called once per integration step, in real time. Data structures for the model inputs and outputs are also created so that information can be passed to and from the model.

The wrapper C code contains the basic framework for real-time operation. This includes the top level routine and timing, as well as the low level functions for hardware interfacing. It is written as a single thread with interrupts for the clock and the receipt of messages. The wrapper code was initially developed by Leonas Bernotas for other applications at MIT Lincoln Laboratory and was adapted for this application.

The wrapper code is compiled using Code Composer Studio (CCS). This product is provided by Texas Instruments as part of the development tools. It is an optimizing C compiler for this specific DSP target. Small pieces of assembly code are required to implement interrupts and boot operation. The memory configuration and control of specific storage locations is managed using additional commands to the linker.

4.3 Efficiency Techniques

The automated process for software implementation produces code that is inefficient. For this particular application, the computation requirements are significant and the inefficient code is unacceptable. This problem has been remedied using a variety of software techniques.

The process began by optimizing the memory map so that critical data was stored in fast memory. This is described in Section 4.3.1. This improved execution speed, but also motivated a reduction in memory usage.

Further improvements were made by optimizing the computational techniques. Data storage and matrix operations were streamlined using custom algorithms. The process of automatic code generation was preserved in anticipation of future development of the Simulink model. Computational complexity was also reduced by simplifying the system model. Reductions in the number of states and measurements were made.

These methods ultimately led to the successful implementation of the filter on hardware at an execution rate of 100 Hz, which matched the IMU message rate. However, several sacrifices were made that ultimately led to degradation in performance.

4.3.1 Memory Map

The filter algorithm requires the storage of large data variables, such as the covariance matrix of the estimation error. Significant increases in execution speed were obtained by optimizing the location of data storage.

The memory locations that are available for data storage are summarized in Table 4-1. The speed of the memory varies inversely to the size of the memory. Speed can be increased by placing critical components in fast memory, but also by reducing memory requirements so that the fastest memory can be used.

Table 4-1: Data Memory Comparison

Memory	Location	Size*	Speed
Internal Memory	on chip	64 KB	Fast
SBSRAM	on board	896 KB	Medium
DRAM	on board	8 MB	Slow

* program storage has been excluded

Ultimately, it was desired to place all model data in the Internal Memory. This memory is extremely fast, because it is on chip with a direct connection to the CPU. However, this memory is limited to 64 KB, which was too small for the automatically generated code. This motivated a reduction in memory usage for the filter algorithm, which was accomplished using several computational techniques that are described in the following sections.

4.3.2 Custom Matrix Operation Routines

The Kalman filter contains several matrix operations for covariance propagation and gain calculations. This computation dominates the CPU usage for a filter of this size. Large increases in the execution rate were accomplished by increasing the efficiency of this computation.

The inefficiency of the RTW code stems from several sources. Intermediate results are stored and copied frequently, including large matrices. Also, the symmetry of certain matrices is ignored because generic algorithms for matrix multiplication are used.

It was undesirable to modify the RTW code directly because this would break the automatic link between Simulink models and C code. This would make future updates more difficult, and lead to differences between simulation and hardware implementation. Instead, algorithm modifications were made using a Simulink S-function. An S-function is a special block that calls C code from the Simulink model.

Using an S-function, I rewrote the entire Kalman Update block in C. In this code, I incorporated several techniques for reducing the memory and computation requirements. I developed these techniques without reference, although matrix processing is a common topic that has undoubtedly been explored by others.

Matrix Indexing and Storage

The revised Kalman Update block achieves efficiency using several techniques. The first technique, described in this section, is an efficient method of data storage for symmetric and diagonal matrices.

All matrices are ultimately stored as one-dimensional data arrays. Simulink and RTW store matrices in their entirety, requiring n^2 elements for an n by n matrix. Indexing is arranged in "column-major" order, as shown at the left of Figure 4-6.

$$\begin{bmatrix} 0 & 4 & 8 & 12 \\ 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \end{bmatrix} \quad \begin{bmatrix} 0 & - & - & - \\ 1 & 4 & - & - \\ 2 & 5 & 7 & - \\ 3 & 6 & 8 & 9 \end{bmatrix} \quad \begin{bmatrix} 0 & - & - & - \\ - & 1 & - & - \\ - & - & 2 & - \\ - & - & - & 3 \end{bmatrix}$$

Figure 4-6: Indexing for Non-symmetric, Symmetric, and Diagonal Matrices

Zero-based matrix indexing can be done using the following formula:

$$A(i, j) = a\left[j * m + i\right], \tag{4-1}$$

where m is the number of rows in A (for square matrices $m = n$) and a is the one dimensional array for data storage. Further as shown in Figure 4-6, this technique can be improved for symmetric matrices by omitting storage of the upper triangle. This reduces the number of elements to $n(n + 1)/2$. Matrix indexing in this symmetric form can be done using the following formula:

$$P(i, j) = p\left[j * n - \frac{j(j+1)}{2} + i \right] \quad \text{for i} \geq \text{j}. \tag{4-2}$$

If $j > i$, they should be swapped before this formula is applied. Note that $j(j+1)$ is always even, so the index is always an integer. This formula was derived using the sum of an arithmetic series for columns before j.

Similarly, diagonal matrices can be stored and indexed by their diagonal:

$$D(i,i) = d[i]. \tag{4-3}$$

Matrix Transpose through Indexing

A second significant inefficiency is the implementation of the matrix transpose. With RTW, the data in a matrix is physically moved each time the matrix is transposed. For symmetric matrices, the transpose operation is entirely unnecessary. For non-symmetric matrices, the transpose can be accomplished by reversing the indices instead:

$$A^T(i, j) = A(j, i), \tag{4-4}$$

leaving the memory array unchanged. This eliminates both the work of transposing and the memory required to store the transpose.

One Pass Multiplication

A final inefficiency was the need to store intermediate results of a large matrix formula. Consider the covariance propagation equation:

$$\dot{P}_\lambda = (A + \lambda I)P + P(A + \lambda I)^T + Q. \tag{4-5}$$

The RTW implementation requires several separate matrix multiplications and additions, with the storage of the intermediate results for each step. Instead, I programmed this entire equation in one pass using the following formula for each element:

$$\dot{P}_\lambda(i, j) = 2\lambda P(i, j) + Q(i, j) + \sum_{k=1}^{n} \left(A(i, k)P(k, j) + P(i, k)A(j, k) \right). \tag{4-6}$$

The matrix multiplications AP and PA^T are performed together in the sum, using swapped indices for A^T. The intermediate results are all scalars, so the input and output are the only data arrays that need storage. Similar efficiency is available for the discrete measurement update.

4.3.3 Processing Measurement One at a Time

One-at-a-time measurement processing has also been employed. The algorithm for this technique was described in Section 2.2.5.3. This method eliminates the matrix inversion in the Kalman gain calculation. This algorithm is natural to program in C inside the S-function for the Kalman Filter.

A result of this algorithm is that the execution time of a measurement update is proportional to the number of measurements in the update. This allows for faster updates in between GPS measurements. The timing algorithm was modified to allow sharing of time between cycles, so that GPS updates could borrow a few milliseconds from the other updates.

A consequence of one-at-a-time measurement processing is that the measurement noise covariance must be diagonal. This limits the noise modeling of the GPS, which usually has poorer accuracy in the vertical direction.

It is likely that one-at-a-time measurement processing degraded numerical performance. The parasitic effects of rounding would be greater in a large set of serial scalar operations than in a single set of matrix operations.

4.3.4 Reduction of Model Complexity

In addition to the previous techniques for computational efficiency, it has been necessary to reduce model complexity. Some of these changes degrade filter performance. This should serve as motivation for further increasing the hardware and software performance so that these sacrifices can be avoided.

All bias states have been removed from the filter for hardware implementation. This includes a total of 9 states for biases in the gyroscopes, accelerometers, and gravity model. This provides a huge reduction in matrix memory and multiplication operations. As studied in Section 3.4.4, the filter can continue to perform well when the omitted biases are small.

The freefall measurement has also been removed. This measurement is not useful for a benchtop test.

Finally, the differential equation solver has been reduced to a linear solver. Here, the Euler approximation

$$\dot{x}(t) \approx \frac{x(t) - x(t-T)}{T} \tag{4-7}$$

is used to convert the continuous differential equations to a discrete system. The execution time of the solution is approximately proportional to the order of the solver, so this provided a substantial improvement in computation speed. However, the linear solver is prone to numerical instabilities during rapid changes in the solution, as discussed in the General Observations portion of Section 4.4.

4.4 Results

The hardware has been assembled and programmed and the performance has been assessed. For this benchtop prototype, it was not possible to test all filter functions, but several metrics of performance have been established.

The general setup is shown in Figure 4-7. Throughout these tests, the GPS antenna (not shown) is mounted in a fixed position. The IMU is free to move because it is attached using a flexible cable. A precision right angle block is used to provide several reference surfaces.

Figure 4-7: Hardware Setup with Right Angle Block

The test procedures and results are described in the following sections. They include a test of attitude accuracy, a limited test of positioning, and a test of the nonlinear initialization routine. The results show reasonable performance, although several issues are noted.

General Observations

The most noticeable weakness is in the robustness of the computation. Under various circumstances, the filter covariance can become indefinite. This produces negative variance and ultimately leads to a numerical failure. Following this, the hardware must be restarted and reinitialized to continue operation.

These problems do not occur in simulation and are the result of differences in the hardware implementation of the algorithm. Several of the implementation techniques, designed to improve execution speed, have compromised the filter performance.

The most significant factor is the use of the linear ODE solver. This is especially noticeable at filter startup, when large convergence of the covariance occurs in the first iteration. It is also prevalent in the attitude covariance propagation during fast rotations. Recall that the attitude accuracy tends to exhibit poor accuracy in one direction, which is usually heading. In body coordinates, this direction of worst accuracy changes when the body rotates. For this reason, certain axes of attitude covariance converge quickly during body rotations.

Additional numerical problems have been created by the use of one-at-a-time measurement processing. Because measurements are processed in series, rounding errors are compounded. However, this effect is small and generally acceptable.

These effects have made filter tuning substantially difficult. Many parameter sets that performed satisfactorily in simulation were numerical unstable on the hardware. As a result, suboptimal filter noise parameters were chosen. The initial covariance was reduced in order

to eliminate large corrections from the first iteration. Similarly, measurement noise covariance was increased above the actual sensor characteristics so that measurement corrections were less abrupt. Process noise was also increased so that estimate uncertainty remained larger. However, the use of large process noise for the force and torque disturbance could also produce numerical failures.

These tuning methods compromise filter accuracy but produce stability. They are quick fixes to a more complicated problem. Several techniques are available for eliminating these hardware numerical problems. These solutions are presented in the recommendations of Section 6.

Attitude with Right Angle Block

This test measures attitude accuracy by moving the IMU through a set of motions. The IMU is moved by hand, so there is no precise control, but a precision right angle block is used to provide several reference surfaces. The pure INS solution is also computed as a reference. The sequence of motions is described in Table 4-2. Time periods are given when the base of the IMU is flush with the various surfaces of the reference block. This constrains motion to body roll. The first cube face is revisited to provide a measure of repeatability.

Table 4-2: Test Motion Sequence

Event	Time (s)	Description
1	11-16	Stationary on face 1
2	24-29	Stationary on face 2
3	34-43	Stationary on face 3
4	49-56	Stationary on face 1
5	56-64	Pure roll on face 1

Table 4-3: Angle between Body x Axis

Events		KF	INS
1	2	90.034	90.0230
1	3	90.028	90.030
1	4*	0.0455	0.0452
2	3	90.009	90.031
2	4	90.009	89.997
3	4	89.999	89.993

* Note: Event 4 was the return to face 1

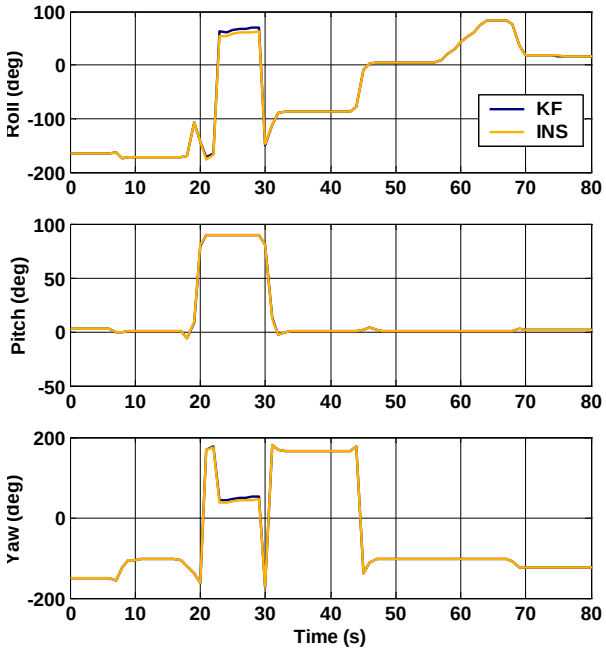

Figure 4-8: Attitude Comparison

The attitude solution during this motion has been converted to NED Euler angles and is plotted in Figure 4-8. For comparison, both the Kalman filter (KF) and INS solution are shown, although there is very little discrepancy between these two estimates. The noticeable

disagreement in roll and yaw during Event 2 is due to the pitch angle near 90º. Roll and yaw are ambiguous at this orientation.

The angles between the body *x* axis at different reference orientations have been calculated in Table 4-3. The values show a maximum disagreement between the filter and reference cube of 0.0455º, which occurred between successive visits to block face 1. Note that small dust particles on the block surface may have produced this result, so it is possible that the underlying accuracy is better. The maximum disagreement between the Filter and INS solution is 0.022º.

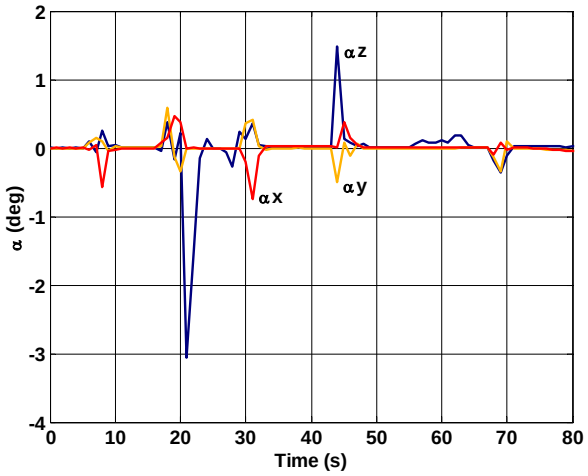

Figure 4-9: Error between Filter and INS

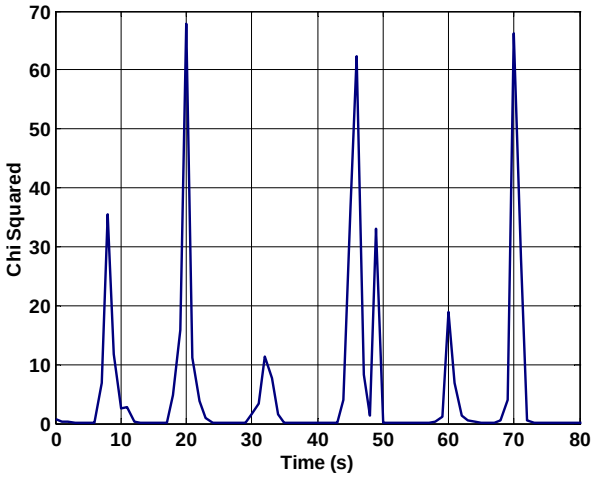

Figure 4-10: Chi-Squared Statistic

The error between the Kalman filter and INS is plotted in Figure 4-9. The error is small when the IMU is stationary but large errors develop during IMU motion. This is undesirable and results from a lag in the Kalman filter solution.

This behavior can be understood using the chi-squared statistic, shown in Figure 4-10. During each period of rotation, the chi-squared value spikes just before the periods of attitude error. These spikes arise because the torque disturbance, which is produced by manually rotating the IMU, is larger than the disturbance noise model. As a result, the filter is forced to compromise between the gyroscope measurement and the disturbance model. The result is a delayed response.

In theory, this type of error can be eliminated by modifying the torque disturbance process model. Both the process noise and the decay rate should be increased. This would allow the filter to give full credibility to the gyroscope measurement under abrupt motion. Unfortunately, the numerical instabilities of the hardware implementation prevented proper tuning. This behavior is the direct result of the suboptimal tuning.

Inertial Positioning During GPS Dropout

Positioning is difficult to test in the benchtop test environment because the GPS antenna is mounted at a fixed location. However, small motion of the IMU is possible. IMU motion without GPS motion is an unrealistic situation for the filter. For this reason, inertial positioning is tested during a GPS dropout.

This test was conducted as follows. First, the GPS signal measurements were removed to simulate a dropout. Then the IMU was moved vertically in a square wave motion of approximately 0.5 meters. Afterwards, the GPS measurement was restored.

The result is plotted in Figure 4-11. The step motions in attitude are readily apparent. Note that the altitude estimate begins to drift downwards during the dropout, but is corrected immediately when the GPS solution returns. This is caused by the integration of noise in the accelerometers and is the natural drift of any INS solution.

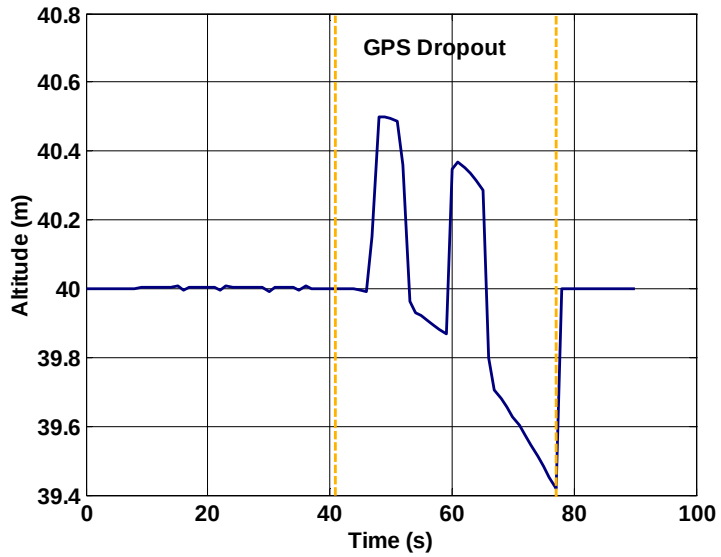

Figure 4-11: Inertial Positioning

A similar IMU motion was also applied before the GPS dropout. In this case, the filter estimate relies heavily on the GPS values. Slight ripples are present because of the measured IMU acceleration, but the altitude estimate remains almost constant.

Large Error Initialization

The performance of the initializer has also been evaluated. For this test, the IMU is positioned in an arbitrary orientation and the navigation system is initialized to the Euler Angles (0, 0, 0). The Earth stationary measurement is enabled, so that the initializer uses the gyrocompassing routine.

The attitude response is shown in Figure 4-12. The initialization error for this test is very large. The gravity alignment quickly orients the roll and pitch axes. Note the over-the-pole motion in pitch, because this was the most direct path for correction. The Earth rate alignment begins at t = 20 seconds, and slowly corrects the yaw alignment. The actual orientation is not precisely known, but this solution appears to be correct. The yaw orientation continues to wander over several degrees because of the sensitivity to gyroscope noise. The resulting solution is sufficiently accurate for the Extended Kalman Filter to converge and a successful handoff was demonstrated.

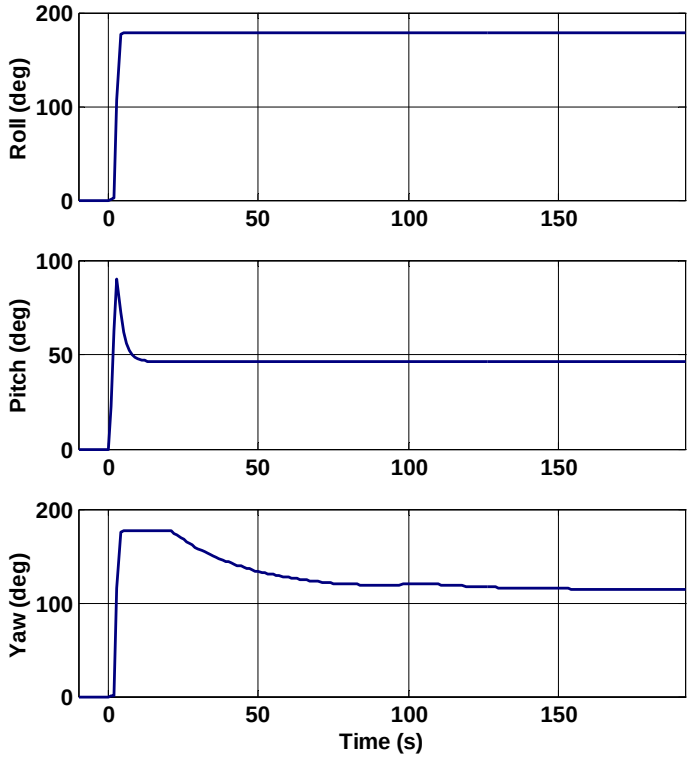

Figure 4-12: Large Angle Initialization (NED)

This concludes the hardware section of this thesis. We now proceed to discussion.

5. Discussion

There were important points and results throughout this design and analysis. In this section, the central issues are revisited and discussed. Opinions are presents on the various decisions of this design.

The use of the Kalman Filter has proven to be successful for this navigation system. The framework of system and noise modeling is useful for addressing and managing the problem. The number of tuning parameters in the filter is slightly large for a system of this order, but this allows a high degree of adjustment.

The concept of data forgetting is very useful because it provides a single parameter for tuning filter response. This allows most noise parameters to be set to their actual value without further adjustment. Many of the covariances can be estimated directly using statistical methods from the available measurement data. Data forgetting can be implemented using a simple modification to the covariance propagation. It can be highly beneficial when modeling errors or measurement glitches are present.

The Extended Kalman Filter (EKF), using the estimated state as the point of linearization, is an invaluable tool for nonlinear systems. Although this raises some stability concerns, the resulting filter for this navigation system behaves well in this respect. Performance in this application is good, and very nearly optimal.

The strategy throughout this project has been to make the system as linear as possible for the EKF. This began with the selection of state variables, including small angle attitude errors and ECI position and velocity. Measurements were pre-compensated so that measurement nonlinearity could be reduced. The calculation of additional navigation outputs, such as geodetic coordinates, was done separately from the EKF. This approach is highly recommended, and may have contributed to the stability of the EKF.

One real problem with the EKF was noticed during fixed to ground operation: richness in the estimated trajectory is interpreted as richness in the actual trajectory. This problem is apparent only in states that are difficult to observe, such as heading alignment. The result is that the covariance matrix and the Kalman gain become too small. Data forgetting or increased process noise can provide an approximate remedy. The crux of the problem, however, is that the point of linearization is influenced by noise.

A peculiarity of the Kalman filter which compounds this effect is that the covariance estimate is open-loop. Using the initial condition and the system model, the covariance is propagated blindly through time independent of the actual filter error. The measurement update reduces the covariance according to the information content of the measurement, but does not use the measurement itself. The actual measurement value is used only for the update of the state estimate.

Over periods of time, the estimated covariance can deviate from the actual covariance and the resulting Kalman gain will become suboptimal. There is nothing to restore the

covariance to its correct value. Data forgetting improves this problem because it reduces the length of time over which accurate covariance propagation is required. Adaptive tuning techniques, such as [17], may provide improvement in this area.

The computation requirements of this filter are high, especially for hardware implementation. In retrospect, the selection of state variables should carefully compare the computational cost of each state to its benefit in the accuracy and stability of the estimate. Choices for a reduced set of state variables are provided in the recommendations of Section 6.

The selection of states for this filter demonstrates some useful techniques. My method of using small angles that are offloaded to quaternions is completely successful and provides the advantages of increased observability and linearity, while using only three attitude state variables. Another significant advantage is that the small angles are intuitive, and the dynamics of the covariance and estimation error are easy to interpret using these variables. Arguments were made that there is no loss in accuracy by using small angles, since the EKF already requires linearization and the error states are offloaded immediately. Simulations show successful convergence from errors larger than 10 degrees.

The selection of rigid body dynamics as the process is useful because it allows knowledge of the force and torque characteristics to be directly incorporated. The precession of a rotating rigid body adds slightly to the observability. Tolerance to an IMU dropout has been demonstrated. This architecture also allows for an independent measurement of the force and torque, which is useful for a vehicle in freefall.

However, these benefits come at the cost of as many as nine state variables (ω, τ_d, and f_d) in comparison to an IMU-based process model. In applications where the force and torque are difficult to predict, there is limited value in attempting to model the force and torque disturbances. In hardware implementation, the additional states are a significant computational burden. Numerical problems may also arise because the force and torque disturbances have much larger process noise covariance than the other filter states. For these reasons, the use of rigid body dynamics should probably be reserved for exoatmospheric applications.

The use of a loosely-coupled architecture, where GPS position and velocity solutions are treated as measurements, has also proven to be successful. Although a direct comparison to a tight-coupled system was not performed, several comments can be made. The hardware computation limits at this time prohibit real-time implementation of a tightly-coupled system, which would add at least one state and one measurement for each GPS satellite used in the solution. Although there are some theoretical advantages, the actual advantages may be small when considering the available GPS products. It is also unlikely to improve attitude accuracy, because covariance analysis showed that attitude is insensitive to GPS accuracy. However, a tightly-coupled system will eliminate delays in the GPS solution, which may be critical for certain applications.

It was found that observation of attitude depends largely on flight path richness. This is because rotation about the direction of applied force can not be observed at any particular instant. Full observability is achieved by variation in the inertial direction of the force. For

the aircraft and booster flight paths, attitude is observable in all directions because the force varies sufficiently over time. For Earth stationary operation, the filter can continue to operate but accuracy becomes poor in the direction of heading. This can be dramatically improved using the "fixed-to-Earth" measurement.

From the covariance analysis, the attitude error is most sensitive to the accuracy of the gyroscopes and the accelerometers. Gyroscopes are required to propagate the attitude, while accelerometers are used to align the acceleration vector. The GPS accuracy is almost irrelevant here, because large distances are inevitably traveled in inertial coordinates. As a result, campaigns to increase attitude accuracy should focus on improving the IMU.

Bias states can improve filter accuracy, provided that they are observable. Gyroscope biases are observable when attitude is observable and also when the fixed-to-Earth measurement is used. Accelerometer biases are observable in the direction of force. Vehicle (or IMU) rotation can also provide full observability of the accelerometer and gyroscope biases.

The gravity model bias should be omitted because it is barely observable and can be obviated by using a higher order gravity model. This will result in a net savings in computation time and an increase in robustness. There is no need to estimate a variable that can be calculated. For the gravity Jacobian, a spherical model is probably acceptable.

The hardware results show that this algorithm can be implemented successfully in a real-time embedded system. However, it has also revealed that the filter is very challenging computationally. Several techniques have been developed for reducing the computational burden. Still, there is continued motivation for increasing the efficiency of the software and hardware. Robustness must also be improved.

These and other recommendations for future improvements are the subject of the next section of this report.

6. Recommendations

This project has produced several concrete results and tools, but has also suggested some future improvements. In this section, we distinguish between capabilities that are currently available, capabilities that are almost available, and capabilities that should be considered for long-term future development. This provides guidelines for further activities and continued research related to this project.

6.1 Immediate Capabilities

This project provides several capabilities that can be used immediately, without further development effort. These are reviewed here so that they can be utilized.

First, the covariance analysis can provide guidance in the selection of sensors for a navigation system. These results provide a direct relation between the accuracy of the measurement and the accuracy of the state estimate. These results can be used in general, or

additional simulations can be run using the noise characteristics and trajectory of a specific application. In this fashion, the accuracy of a navigation system can be fully accessed before anything is purchased or built.

Second, the full Simulink implementation can be used for the post-processing of flight data. This will require tuning to the specific sensors and disturbance characteristics, but will produce a high quality navigation solution. Indeed, this solution arguably out-performed the commercial solution in the aircraft data set. My navigation solution can be used to processes onboard imaging data or to assess the quality of another navigation solution.

Finally, the software and hardware components of this project can be used as a starting point for future development of navigation systems. Although improvements are warranted, and discussed below, this project provides a large resource toward similar efforts.

6.2 Direct Improvements

There are several additional capabilities that can be achieved with minimal further effort in algorithm design. Pursuing these will greatly extend the filter performance.

Hardware Execution Speed

Hardware implementation was difficult because of the complexity of this algorithm. Improvements were made to allow for successful real-time performance, but several sacrifices resulted. An increase in the speed of the hardware would eliminate these sacrifices and could allow other improvements in the filter performance.

Figure 6-1 shows the dimensions of performance. All of these axes can be increased with minimal modifications to the algorithm but they require additional computation time. Improvement in execution speed can therefore provide direct benefits in these areas.

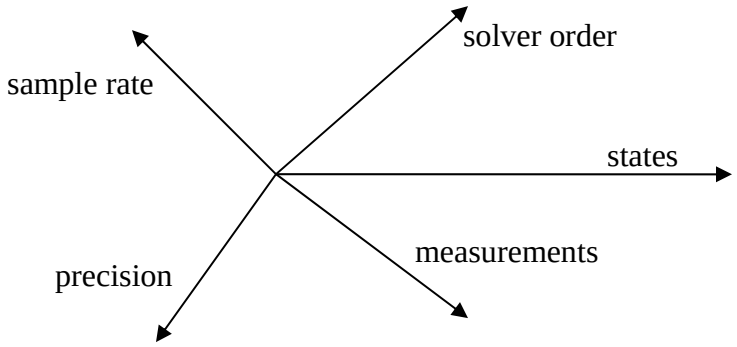

Figure 6-1: Dimensions of Performance

These dimensions include the number of states and measurements of the filter. For this hardware implementation, it was necessary to remove the bias states and the freefall

measurement. Pursuit of a tightly-coupled solution would require additional states and variables. These new variables would increase the size of all matrices, impacting memory and multiplication operations.

Increasing the solver order can dramatically improve the numerical robustness of the filter. The use of a linear solver was by far the gravest sacrifice that was made when porting to hardware. Execution time is approximately proportional to solver order, although the measurement updates occur at a constant frequency. In a similar fashion, the accuracy can be improved by increasing the sample rate.

The use of double precision floating point computation could be considered, although this may not be a wise use of resources compared to the other axes of performance. Still, the current single-point precision is producing a reduction in numerical robustness.

Modeling Improvements

This project has explored a large set of state variables. In response to the results of simulations and observability analysis, the choice of state variables should be revisited. Hardware implementation motivates a reduction in state variables. Certain applications may also benefit from a custom choice of state variables.

The modeling of biases should be revisited. The gravity model bias should certainly be eliminated in exchange for a higher-order gravity model. Gyroscope and accelerometer biases should be included if resources permit, because they are usually observable and will improve attitude accuracy. Scale factors will provide improvements for applications with large ranges in IMU measurement values, but will require 6 to 18 additional state variables, depending on the inclusion of cross-correlation terms.

For applications that experience unpredictable disturbances, the IMU measurements should become the process model and replace rigid body dynamics. This will remove 9 state variables and eliminate the need for disturbance modeling. It also eliminates the nonlinearity of the body rate equation. Some reformulation of the filter equations is necessary, but this is primarily a simplification of the filter. This derivation is included in the Appendix.

Finally, all applications will benefit if the attitude error α is defined in ECI coordinates. With this modification, the small rotation is performed before the body rotation:

$$B(q) = B(\hat{q})R_{\alpha}\,. \tag{6-1}$$

The benefit is a more linear propagation equation:

$$\dot{\alpha} = B(\hat{q})^{T}\,\delta\omega\,. \tag{6-2}$$

In particular, this choice of variables has eliminates the term $\hat{\omega}\times\alpha$ that appeared when α was defined in body coordinates. ECI attitude error remains constant when the vehicle

rotates. As a result, the propagation of α in ECI coordinates is more robust at high rotation rates.

The measurement update for the quaternions must also be modified slightly to include the transformation into body coordinates:

$$\hat{q}_+ = \hat{q}_- + f_q(\hat{q}_-)B(\hat{q})i_\alpha . \qquad (6\text{-}3)$$

All expressions involving α in the state and measurement Jacobians must be reformulated using (6-1). The IMU process model that is presented in the Appendix includes this modification.

Flight Robustness

The robustness of the filter for flight applications can be improved using several system-level techniques. This is required before flight operation is attempted and will increase filter accuracy in many situations.

The first technique is measurement rejection based on the chi-squared statistic. This can be used to eliminate isolated measurement glitches and prevent the filter from making large corrections. This technique is common practice in most commercial navigation systems.

Second, the INS solution, which is very robust, should be available as a backup solution. The INS solution could benefit from periodic corrections from the Kalman filter. These corrections must be applied only when the Kalman filter is providing a good estimate, which is measured by the covariance and chi-squared statistic. Corrections could include attitude and position updates, as well as bias information. This would provide the ability to restart the Kalman filter in the event that divergence occurs.

Finally, the nonlinear initializer for the Kalman filter should be expanded into a full nonlinear estimator and used whenever the error covariance is large. This would include initialization, but also periods of poor observability such as sensor dropouts. In this fashion, the Kalman filter would be completely protected from nonlinear effects. The nonlinear estimator would only require coarse accuracy within the capture range of the Kalman filter, and could ignore biases and small effects. The optimal linear solution of the Kalman filter would provide higher accuracy when the state errors are small. The main task required for this improvement is the development of an attitude alignment procedure that does not require the vehicle to be Earth stationary. This should be achievable by fitting the INS solution to the GPS data.

6.3 Future Objectives

Additional improvements have been identified that require future research. These topics have been motivated by this project, but were not pursued in this thesis.

Square-Root Filtering

Difficulties in the hardware implementation arose because the covariance matrix could become indefinite under imperfect arithmetic and low order ODE solvers. The use of the square-root filtering algorithm would greatly improve numerical performance by enforcing the positive definiteness of the covariance matrix.

This routine was described in Section 2.2.5.3. It can directly replace the current Kalman filter block in the Simulink model. It is expected that computation time will increase but accuracy and robustness will improve. The most significant benefits will be in the hardware performance.

Tightly-Coupled Architecture

Although many arguments were presented against it, and the current hardware prohibits it, the tightly coupled filter architecture has merit and can lead to improved accuracy. The greatest reason for using it is the elimination of delays and dynamics in the GPS solution. A "head to head" simulation comparison of the tightly-coupled and loosely-coupled architectures would provide interesting results, regardless of the victor.

Implementation of a tightly-coupled navigation system in the simulation environment will require several tasks. The first is rudimentary processing of the GPS messages to obtain pseudorange and satellite position from the raw messages. The accuracy of this pseudorange measurement can be improved using more advanced techniques, such as phase measurement and dual frequency comparison. Differential GPS and additional augmentation systems must be incorporated at the pseudorange level, before the GPS position solution is calculated. The pseudorange measurements should then replace the GPS position and velocity measurements in the navigation filter. The Doppler measurement can be used as a measurement of the derivative of pseudorange.

Implementation on hardware will require significant improvements in computational speed or algorithm efficiency. Simulation will reveal if it is worthwhile to purse these efforts.

Forward-Backward Smoothing

For post-processing applications, the filter accuracy can be improved by incorporating data smoothing. In these methods, the optimal estimate is calculated using the entire measurement set, including future measurements. A popular approach is known as Forward-

Backward smoothing and is described in [3, pp. 322-331]. This technique involves building two Kalman filters, running one forward in time and one backward in time. The optimal (minimum variance) combination of these two estimates is the variance weighted average:

$$\hat{x} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\hat{x}_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\hat{x}_2 . \tag{6-4}$$

The estimate with the smaller variance is given more weight. This provides a new estimate with variance

$$\sigma^2 = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}, \tag{6-5}$$

which is smaller than the variances each individual estimate. Implementing this technique requires modified equations for the backward filter, but can use the same Jacobians and system model. Computation time will nearly double, because the covariance propagation and Kalman update dominate, but the required implementation effort is minimal.

This technique is certainly useless for real-time operation, so hardware implementation is not an issue.

Additional Measurements

Problems in state observability can usually be resolved by taking measurements from additional sensors. This also can improve accuracy for states that are already observable. There are several leading candidates for this pursuit.

Of the critical navigation states, attitude exhibited the worst observability, especially in the direction of heading. Heading accuracy can be measured using a magnetic compass. This will require a magnetic Earth model and a corresponding error model. Although a compass measurement is not precise, it can provide benefit because it does not usually drift over time.

A more accurate and expensive attitude measurement is a star tracker, which measures full attitude by sighting a celestial object or manmade beacon. This sensor typically requires a motion controlled telescope and camera with the ability to track a light source.

Another alternative is a phased array receiver which measures attitude by the phase differences in the received signals. This can be performed using the signals from a land based transmitter or from the GPS satellites. As with all phase measurements, there is ambiguity between successive wave periods which must be resolved.

Mixed Continuous and Discrete States

While continuous differential equations were ideal for representing the dynamics of motion, they were awkward and inefficient for representing sensor noise. The biases themselves are

apparent only at the discrete times of measurement, so it would be more natural to model them as discrete time states. This would also permit the direct use of a number of powerful discrete system modeling tools.

A suggestion is to develop the minimum variance filter for a system that has both discrete and continuous states. The measurement update could remain unchanged, but the covariance propagation would need to be modified. The result would allow for the modeling of continuous and discrete states where they are most appropriate. This would ultimately lead to fewer continuous states, which would probably reduce computation time.

This completes the recommendations for this project.

7. Conclusion

This thesis presented the theory, simulation, and hardware prototype for a navigation system. IMU and GPS measurements were fit to rigid body dynamics using an Extended Kalman Filter.

The theoretical derivation presented the details for constructing this filter. Special attention was given to the assumptions and optimality conditions of the Kalman Filter. Observability of the system was studied in detail and statements concerning stability and convergence were made.

Simulation provided characterization of the filter performance. Covariance analysis determined the accuracy of the filter and sensitivity to different parameters. Full simulation revealed filter behavior in both normal and abnormal operating conditions. Flight simulations validated the filter using real flight data.

Hardware implementation demonstrated that this algorithm could be run in real time. The computational burden was significant, but improvements in algorithm efficiency and computation speed were achieved using several methods. However, the robustness and accuracy of the hardware prototype were ultimately compromised.

This project achieved successful performance in both simulation and hardware. It provides a source of information and insight for the design of navigation systems. This thesis stands as a foundation for future development in this research area at MIT Lincoln Laboratory.

Appendix – IMU Process Model

A process model based on the IMU measurements is defined here and the resulting linearization for the EKF is derived. This reduces the number of filter states and measurements, and is more appropriate for systems with unpredictable dynamics. This analysis is brief, but it uses the same techniques that were used in Section 2.3.1 for the linearization of rigid body dynamics.

The IMU measurement model

$$\omega_m = \omega + b_\omega + n_\omega \tag{A-1}$$

$$a_m = B(q)\left(\dot{v} - g(p)\right) + b_a + n_a \tag{A-2}$$

can be inverted to obtain the following process model:

$$\omega = \omega_m - b_\omega - w_\omega \tag{A-3}$$

$$\dot{v} = B(q)^T\left(a_m - b_a\right) + g(p) - B(q)^T w_a\,. \tag{A-4}$$

The IMU measurements are now treated as known control inputs instead of measurements. A zero order hold must be applied so that the measurement is available in continuous time. Note that the discrete measurement noise *n* has become the continuous process noise *w*, so the value must be adjusted according to the sample time. Specifically

$$R = TQ\,, \tag{A-5}$$

where *T* is the sample time, *Q* is the discrete-time covariance matrix:

$$E\left\{nn^T\right\} = Q\,, \tag{A-6}$$

and *R* is the continuous-time covariance matrix:

$$E\left\{ww^T\right\} = R\delta(t)\,. \tag{A-7}$$

This result can be derived by equating the effects of continuous-time and discrete-time noise over one sample period.

Nonlinear Dynamics

The propagation for the fundamental navigation states becomes

$$\dot{q} = f_q(q)\left(\omega_m - b_\omega - w_\omega\right) \tag{A-8}$$

$$\dot{p} = v \tag{A-9}$$

$$\dot{v} = B(q)^T\left(a_m - b_a\right)B(q)^T + g(p) - B(q)^T w_a \tag{A-10}$$

Note that ω is not propagated in this process model. Bias states remain modeled as simple decays with $\dot{x}_i = -r_i x_i$.

Linearized Small Error Dynamics

The linearization about the estimated state is now

$$\dot{\alpha} = -B(\hat{q})^T \delta b_\omega - B(\hat{q})^T w_\omega \tag{A-11}$$

$$\delta\dot{p} = \delta v \tag{A-12}$$

$$\delta\dot{v} = B(\hat{q})^T [(\hat{b}_a - a_m)\times]\alpha - G_p \delta p - B(\hat{q})^T \delta b_a - B(q)^T w_a, \tag{A-13}$$

The attitude error α has been redefined in ECI coordinates, which eliminates ω_m from the error propagation. Bias state errors propagate according to $\delta\dot{x}_i = -r_i \delta x_i$. The resulting linear matrix expression $\delta\dot{x} = A\delta x + Fw$ is

$$\begin{pmatrix} \dot{\alpha} \\ \delta\dot{p} \\ \delta\dot{v} \\ \delta\dot{b}_\omega \\ \delta\dot{b}_a \end{pmatrix} \approx \begin{bmatrix} 0 & 0 & 0 & -\hat{B}^T & 0 \\ 0 & 0 & I & 0 & 0 \\ A_{v\alpha} & G_p & 0 & 0 & -\hat{B}^T \\ 0 & 0 & 0 & -r_{b\omega} I & 0 \\ 0 & 0 & 0 & 0 & -r_{ba} I \end{bmatrix} \begin{pmatrix} \alpha \\ \delta p \\ \delta v \\ \delta b_\omega \\ \delta b_a \end{pmatrix} + \begin{bmatrix} -\hat{B}^T & 0 \\ 0 & 0 \\ 0 & -\hat{B}^T \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} n_\omega \\ n_a \end{pmatrix}, \tag{A-14}$$

where $A_{v\alpha} = \hat{B}^T [(\hat{b}_a - a_m)\times]$. The rotation $-\hat{B}^T$ in F is unimportant if the IMU noise is uncoupled and identical in all axes, i.e. $E\{nn^T\} = \sigma^2 I$, because then

$$E\left\{\left(-\hat{B}^T n\right)\left(-\hat{B}^T n\right)^T\right\} = \sigma^2 \hat{B}^T \hat{B} = \sigma^2. \tag{A-15}$$

Measurement Compensation and Linearization

The number of measurements is reduced because the IMU is no longer treated as a measurement. In addition, the freefall measurement is only applicable to the translational equations. The remaining measurements can be compensates as follows, producing small error measurements:

$$\delta\omega_{fix} = -B(\hat{q})^T (\omega_m - b_w) - \omega_e \tag{A-16}$$

$$\delta v_{fix} = \omega_e \times R_e \hat{p} - R_e \hat{v} \tag{A-17}$$

$$\delta a_{free} = -a_m - \hat{b}_a \tag{A-18}$$

$$\delta p_{gps} = p_{gps} - R_e \hat{p} \tag{A-19}$$

$$\delta v_{gps} = v_{gps} + \omega_e \times R_e \hat{p} - R_e \hat{v}, \tag{A-20}$$

The resulting linear measurement equation $\delta z = H\delta x + n$ is

$$\begin{pmatrix} \delta\omega_{fix} \\ \delta v_{fix} \\ \delta a_{free} \\ \delta p_{gps} \\ \delta v_{gps} \end{pmatrix} = \begin{bmatrix} \hat{B}\left(\hat{b}_\omega - \omega_m\right) & 0 & 0 & -\hat{B} & 0 \\ 0 & -\omega_e \times R_e & R_e & 0 & 0 \\ 0 & 0 & 0 & 0 & -I \\ 0 & R_e & 0 & 0 & 0 \\ 0 & -\omega_e \times R_e & R_e & 0 & 0 \end{bmatrix} \begin{pmatrix} \alpha \\ \delta p \\ \delta v \\ \delta b_\omega \\ \delta b_a \end{pmatrix} + \begin{pmatrix} n_{\omega fix} + n_\omega \\ n_{vfix} \\ n_{free} + n_a \\ n_{pgps} \\ n_{vgps} \end{pmatrix}. \quad \text{(A-21)}$$

The noise terms n_{vfix}, n_{pgps}, and n_{vgps} are the respective measurement noises. However, the other measurements contain a combination of noise sources. The measurement ω_{fix} contains noise from the gyroscopes and from vehicle vibrations. Similarly, the measurement a_{free} contains noise from the accelerometers and from disturbance forces. Assuming independence, the covariance of the sum is equal to the sum of the covariances for these two noise sources.

Error Offloading

All error states should be immediately offloaded to the nonlinear state estimate. With α in ECI coordinates, the attitude update is now

$$\hat{q}_+ = \hat{q}_- + f_q(\hat{q}_-)B(\hat{q})i_\alpha . \quad \text{(A-22)}$$

Observability

The observability results are very similar to the rigid body model in most respects. The definition of the attitude error α in ECI coordinates simplifies analysis.

Attitude alignment is weak about the axis of measured acceleration, which is equivalent to the direction of applied force. Attitude becomes fully observable when the direction of measured acceleration is varied with respect to inertial coordinates.

Gyroscope biases are observable when attitude is fully observable or when the fixed-to-Earth measurement is used. Accelerometer biases are observable in the direction of measured acceleration. Both of these biases become fully observable when the vehicle rotates.

The notable difference is that rigid body dynamics no longer contribute to the observability. This includes the dynamic properties of the force and torque disturbances as well as the precession of a rotating vehicle. However, when these effects are unpredictable, there is very little information lost by using the IMU process model.

References

1. Anderson, B. D. O., "Exponential Data Weighting in the Kalman-Bucy Filter," *Information Sci*, Vol. 5 (1973), pp. 217-230.
2. Anderson, B. D. O., "Stability Properties of Kalman-Bucy Filters", *Journal of the Franklin Institute*, No. 2 (1971), pp. 137-144.
3. Brown, Robert Grover and Hwang, Patrick Y. C., *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley & Sons, New York, 1997.
4. Faruqi, Farhan A., and Turner, Kenneth J., "Extended Kalman Filter Synthesis for Integrated Global Positioning/Inertial Navigation Systems", *Applied Mathematics and Computation*, Vol. 155 (2000), pp. 213-227.
5. Friedland, Bernard, *Control System Design, an Introduction to State Space Methods*, McGraw-Hill, New York, 1986.
6. Gavrilets, Vladislav, "Avionics Systems Development for Small Unmanned Aircraft", *Thesis (M.S.) Massachusetts Institute of Technology*, Dept. of Aeronautics and Astronautics, 1998.
7. Gelb, Arthur, *Applied Optimal Estimation*, M.I.T. Press, Cambridge, Massachusetts, 1974.
8. Grewal, Weill, and Andrews, *Global Positioning Systems, Inertial Navigation, and Integration*, John Wiley & Sons, New York, 2001.
9. Hua, Cuong, "Gyrocompass Alignment with Base Motions: Results for a 1 nmi/h INS/GPS System", *Navigation*, Vol. 47 No. 2 (2000), pp. 65-74.
10. Jazwinski, Andrew H., *Stochastic Processes and Filtering Theory*, Academic Press, New York, 1970.
11. Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *ASME Journal of Basic Engineering*, March (1960), pp. 35-45.
12. Kalman, R. E. and Bucy, R. S., "New Results in Linear Filtering and Prediction Theory", *ASME Journal of Basic Engineering*, March (1961), pp. 95-108.
13. Lewis, Frank L., *Optimal Estimation with an Introduction to Stochastic Control Theory*, John Wiley & Sons, New York, 1986.
14. National Imagery and Mapping Agency, *Department of Defense World Geodetic System 1984*, Third Edition, 2000.
15. Qi, Honghui and Moore, John B., "Direct Kalman Filtering Approach for GPS/INS Integration", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 38 No. 2 (2002), pp. 687-693.
16. Sage, Andrew P. and Husa, Gary W., "Adaptive Filtering with Unknown Prior Statistic", *Joint Automatic Control Conference of the American Automatic Control Council*, August (1969), pp. 760-769.

17. Sasiadek, J. Z., Wang, Q., and Zaremba, M. B., "INS/GPS Navigation Data Fusion Using Fuzzy Adaptive Kalman Filtering", *IFAC Robot Control* (2000), pp. 423-427.

18. Sofair, Isaac, "Improved Method for Calculating Exact Geodetic Latitude and Altitude", *Journal of Guidance, Control, and Dynamics*, Vol. 20 No. 4 (1997), pp. 824-826.

19. Sofair, Isaac, "Improved Method for Calculating Exact Geodetic Latitude and Altitude Revisited", *Journal of Guidance, Control, and Dynamics*, Vol. 23 No. 2 (2000), p. 369.

20. Stevens, Brian L. and Lewis, Frank L., *Aircraft Control and Simulation*, John Wiley & Sons, New York, 1992.

21. Strang, Gilbert, *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, Cambridge, Massachusetts, 1986.

22. Vik, Bjørnar and Fossen, Thor, "A Nonlinear Observer for GPS and INS Integration", *Proceeding of the 40th IEEE Conference on Decision and Control*, Orlando, Florida, Dec. (2001), pp. 2956-2961.

23. Whittaker, E. T., *A Treatise on the Analytical Dynamics of Particles and Rigid Bodies*, Cambridge University Press, Cambridge, Massachusetts, 1961.