Using A Gimbal To Calibrate An Inertial Measurement Unit

by

Goutam Reddy

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering In Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August 2005

Author . ...
Department of Electrical Engineering and Computer Science
August 22, 2005

Certified by..
Hugh Herr
Professor
:pervisor

Accepted by.................. ......
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Using A Gimbal To Calibrate An Inertial Measurement Unit

by

Goutam Reddy

Submitted to the Department of Electrical Engineering and Computer Science
on August 22, 2005, in partial fulfillment of the
requirements for the degree of
Master of Engineering In Electrical Engineering and Computer Science

Abstract

In this thesis, an inertial measurement unit (IMU) consisting of 3 accelerometers and 3 rate gyros is created using off-the-shelf sensors from STMicro and Analog Devices. A novel technique for calibrating the orientation, position, scaling and offset of each of the sensors on the IMU is developed. A gimbal consisting of three concentric rings, with rotary encoders measuring the rotation between rings is designed. The IMU is fixed to the inner ring of the gimbal and rotated in space. By sweeping appropriate orientations of the IMU at appropriate rates, filtered sensor values can be mapped to "true" angular velocities and linear accelerations computed from the gimbal rotations. The sensor parameters are estimated via. MMSE, and a Kalman filter is implemented to estimate the IMU's attitude (roll and pitch angles) from the raw sensor values. The calibrated sensors are found to track the pitch angle with a mean-square-error of 1.7427 degrees, and the roll angle with a mean-square-error of 3.1387 degrees. The novel outcome of this thesis is that it defines a technique for calibrating IMUs with component sensors that need not be orthogonal in placement.

Thesis Supervisor: Hugh Herr
Title: Professor

Acknowledgments

I would like to acknowledge Professor **Hugh Herr** for providing a guiding hand while still allowing me to forge my own path over the past few years. A special thanks to:

Max Berniker, for helping me hash out some hairy math,

Waleed Farahat, for pointing me in the direction of some hairier math,

Samuel Au, for slipping relevant journal articles onto my desk,

Russ Tedrake, for traversing the land of hairy PC104 drivers,

Dan Paluska, for lending an open ear and some LaTex help,

Bruce Deffenbaugh, for persistently reminding me that ”It’s easy!”,

Ken Pasch, for teaching me little bits of everything.

I would like to thank team **Biomech** (formerly team **Leglab**), and team **Cruftlabs** for the constant creative diversions that keep one from losing it.

And **mom**- for all the warm words and frozen food.

And **dad**- for constantly questioning, “when are you going to finish that paper?”

Knowledge without implementation is time wasted. Implementation without knowledge is effort wasted. May this knowlege save you time and effort in your implementation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 History of Inertial Measurement Units

Inertial Measurements Units are devices that can measure the orientation and position of an object in space. They are generally used to track how an object rotates and translates as it moves through the world. IMUs were popularized in the 1960's as components of inertial navigation systems (INS), which were used by inertial guidance systems (IGS) in missiles and onboard ships, submarines, and space shuttles [1], [13]. They are currently used for mobile robot navigation, user interfaces, and human motion tracking. In the field of mobile robots, IMUs are used in the control of autonomous cars, submarines and unmanned aerial vehicles, [21] They are often augmented with sonar and differential GPS sensors to "retrue" the object's location in space. While general algorithms exist for predicting IMU orientation down to a few degrees resolution, they are prone to error accumulation in the difficult problem of accurate translation tracking. IMUs have also become popular in user interface devices, such as mice, remotes, joysticks, as well as intelligent golf clubs. In these scenarios, the IMU is used more for a coarse measurement of movement, or of actual velocity and acceleration, but not for true orientation and position. Similarly, IMUs are being used in video games interfaces and for gesture recognition, [1].

Of particular interest to the author and his laboratory is the application of IMUs toward human-machine interfaces. A miniature IMU is being developed as a vestibular implant for disabled subjects with balance problems, [25]. A few companies, such as Animazoo and Xsens sell IMU based motion capture systems that outfit the human body and take measurements of human motion and natural and artificial limb trajectories, [35], [17]. Much previous work has been done in using IMUs to sense lower-limb angles as a method for timing biological stimulation (FES), [24]. The author believes that IMUs will continue to play a greater roll in the burgeoning field of intelligent prostheses and orthoses.

1.2 History of Gimbals

A gimbal is a mechanical device that allows for all 3 degrees of rotational freedom. Historically, gimbals have been used for inertial navigation for space shuttles, Figure (1-1), as well as stabilization of missile platforms and boats. In traditional scenarios, an IMU is located in the center of the gimbal, and the gimbal axes are spun by a measured amount to keep the orientation of the IMU with respect to the world the same, [8]. From these rotations at the axes, the Euler angles of the gimbal can be measured.

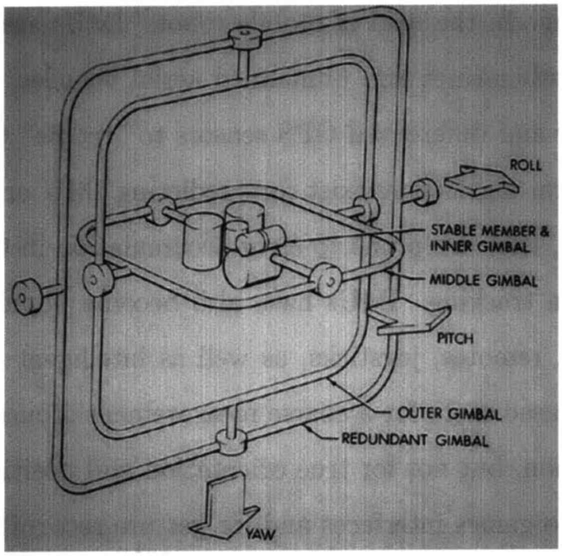

Figure 1-1: 4-axis Gimbal Used on Apollo Space Shuttle [12]

Unfortunately, using Euler angles leads to a problem called Gimbal Lock, in which an axis rotates past 90 degrees, and it looks as though the IMU has flipped inside the gimbal, [13]. The Apollo Space Shuttle missions skirted around the problem of gimbal lock by using a 4-axis gimbal, Figure (1-1). The problem can also be avoided by using a different representation for 3-D rotations such as quaternions, rotation matrices, or axis-angle pairs, [19]. This thesis uses 3-by-3 rotation matrices to transform vectors from one coordinate frame to another, as well as axis-angle (vector) representations of angular rates and accelerations.

1.3 Composition of IMUs

IMUs are traditionally composed of some combination of inclinometers, accelerometers, magnetometers, and rate gyroscopes. Inclinometers measure the inclination of an object with respect to ground (i.e. pitch angle). Accelerometers measure linear acceleration along an axis. Magnetometers measure the magnetic field along an axis. Rate gyroscopes measure the angular velocity about an axis of rotation.

The components come in various packages based on various technologies that have a wide range of associated costs, [1]. Initial technology for rate gyros consisted of a gimballed rotating mass, in which the precession of the object was measured by measuring rotation at gimbal axes. Now, most sensors come in the form factor of an integrated circuit. For example, Analog Devices sells 3 kinds of single-axis MEMS rate gyros for between $22 to $30 a gyro in bulk. Gyration sells a 2-axis rate gyro, while no manufacturer to this point has developed a commercially available 3-axis MEMS rate gyroscope, [28]. Accelerometers come in varieties of piezo, capacitive, and MEMS devices. Analog Devices sells 2-axis 2g accelerometers for $7 a piece. STMicro sells a 3-axis accelerometer with for $22 a chip. Currently, the trend in new technology has been toward developing low cost, low profile MEMS devices. Xsens, Intersense, Xbow, MicroStrain, Silicon Designs are all companies that sell a prepackaged IMU, [34] [30] [31] [33] [29] [32]. While most applications use or assume an IMU created out of orthogonal sensor components, within these multi-axis sensors, there exists an

X-Y skew error of typically up to 2 degrees, [10].

1.4 Previous Work on IMU Calibration

Previous calibration techniques have focused on determining normalizing factors for each of the individual sensors. These factors are usually comprised of a scaling and an offset value for each of the sensors. The orientation and position of the sensors with respect to one another are also estimated in some techniques. These parameters are measured as offsets from 90 degrees because most applications assume an orthogonal, or near-orthogonal layout of sensors, [15].

General calibration techniques involve orienting an accelerometer by hand until a maximum and a minimum is "eye-balled", averaging the two values to achieve an offset, and scaling the range between the max and min to be 2g. Similarly, rate gyros offsets are calibrated by averaging the gyro signal over a period of no motion. Gyro scaling is estimated by rotating the IMU by a certain angle along the "eye-balled" gyro axis in a certain amount of time, and dividing by the integrated gyro signal. Such techniques require apriori information as to the placement of sensors on the IMU, a careful monitoring of an induced motion and timing, and are subject to qualitative approximation, [26].

While previous calibration techniques yield acceptable results for applications that do not rely on precise angular measurement, such as gesture recognition, [1], they are NOT adequate for accurate estimation of yaw, pitch, and roll (YPR) angles. This thesis attempts to fill the gap in available technology for calibrating inertial measurement units which can be assembled by hand, and which can have non-orthogonal sensor alignment.

In particular, at the time of this thesis, commercially available IMUs consisting of 3 accelerometers and 3 rate gyros take on form factors which may not be suited for certain applications. An IMU can be created by combining 1 and 2-axis components, but the orthogonality of the sensors on such a "hand-built" IMU cannot be guaranteed. Furthermore, in the application of a slim-line IMU which conforms to a non-orthogonal shape, an IMU purposefully composed of non-orthogonal sensors may be created. In both these situations, the procedure described in this thesis allows for the IMU to be calibrated.

1.5 Chapter Overview

Chapter 2 defines the problem of IMU calibration and covers, in mathematical detail, the algorithm used to calibrate accelerometer and rate gyro readings from known gimbal rotations. Chapter 3 focuses on the Kalman filter algorithm used to estimate roll, pitch, and yaw from the calibrated IMU. Chapter 4 details the physical hardware implementation of the IMU and the gimbal, as well as the data acquisition and processing hardware used to calibrate the IMU. Chapter 5 characterizes the electrical noise profile of the IMU sensors, and exposes the problem of discretization by rotary encoders in the gimbal. Chapter 6 discusses the filters used to compensate for the noise and discretization. Chapter 7 contains plots of convergence and analysis of error of the calibrated IMU sensors, as well as quantifies the predictive success of the Kalman filter. Chapter 8 outlines improvements in the calibration algorithm and IMU and gimbal construction. Chapter 9 ends with a summary of the results.

Chapter 2

Calibration Mathematics

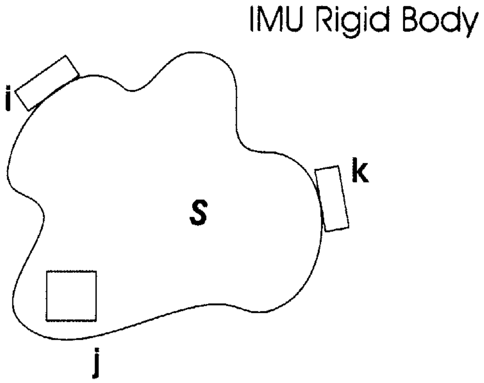

Figure 2-1: An IMU Containing Sensors i,j,k in Sensor Reference Frame, **S**

2.1 Problem Statement

Given an Inertial Measurement Unit (IMU) comprised of accelerometers and rate gyros placed in arbitrary orientations and locations, (Figure 2-1), calibrate the sensors individually, and with respect to one another. Calibration means finding the scaling, offset and orientation of each of the sensors on the IMU, as well as the positions of the accelerometers on the IMU.

2.2 IMU Equations

An IMU is composed of multiple sensors fixed in a sensor reference frame **S**. The IMU rotates and translates in the world reference frame **W**. The accelerometers and rate gyros are fixed in the IMU's sensor reference frame, **S**. Each of the sensors is located at a fixed position and orientation in **S**. Each sensor outputs a voltage between 0V and 5V with some offset voltage corresponding to "zero" for that sensor, and some scaling which converts the voltage to a measurement (e.g. m/sec^2 or rad/sec).

2.2.1 Accelerometer Equations

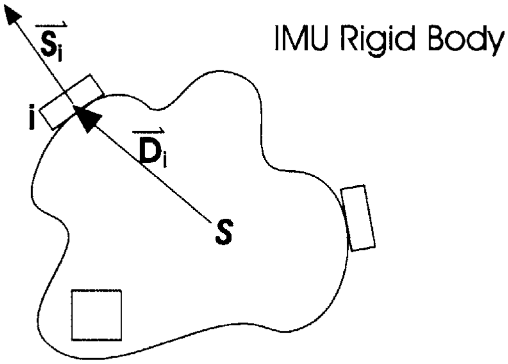

Figure 2-2: Accelerometer i Situated on the IMU in **S**

Accelerometer i is situated in **S**, (Figure 2-2), with

- position vector $\vec{D_{i\mathbf{S}}} = (d_x, d_y, d_z)^T$
- orientation vector $\vec{S_{i\mathbf{S}}} = (s_x, s_y, s_z)^T$
- scale factor $|\vec{S_{i\mathbf{S}}}|$
- offset voltage μ_i
- output voltage A_i

Also note that the scalings which convert voltages to measurements of linear acceleration can be represented as the magnitudes of the orientation vectors for each of the sensors.

An accelerometer i measures a component of the IMU's linear acceleration in the world reference frame, $\vec{l_{\mathbf{W}}}$. In particular, an accelerometer outputs a voltage equal to the projection of $\vec{l_{\mathbf{W}}}$ onto $\vec{S_{i\mathbf{W}}}$, multiplied by some scaling, $|\vec{S_{i\mathbf{W}}}|$, plus some offset voltage, μ_i.

$$A_i = |\vec{S_{i\mathbf{W}}}| \cdot Proj_{\vec{S_{i\mathbf{W}}}}(\vec{l_{\mathbf{W}}}) + \mu_i \tag{2.1}$$

Equation (2.1) can be further simplified to

$$A_i = \vec{S_{i\mathbf{W}}} \circ \vec{l_{\mathbf{W}}} + \mu_i \tag{2.2}$$

Because $\mathbf{S}$ and $\mathbf{W}$ are non-inertial reference frames (i.e. the IMU can accelerate and rotate in $\mathbf{W}$), $\vec{l_{\mathbf{W}}}$ is derived from translational and rotational components. The translational acceleration is composed of a static gravity component, $\vec{G} = (0, 0, -9.81)^T$, and a dynamic component $\vec{h}$ which represents the IMU's linear acceleration due to translational motion. The rotational acceleration stems from the angular acceleration, $\vec{\alpha_{\mathbf{W}}}$, and the centripetal acceleration, $\vec{C_{\alpha\mathbf{W}}}$, and acts along the sensor's position vector rotated into the world frame, $\vec{D_{i\mathbf{W}}}$.

$$\vec{l_{\mathbf{W}}} = \vec{G} + \vec{h} + (\vec{\alpha_{\mathbf{W}}} \times \vec{D_{i\mathbf{W}}}) + \vec{C_{\alpha\mathbf{W}}} \tag{2.3}$$

The centripetal acceleration can be described in terms of the angular velocity of the IMU in the world reference frame, $\vec{\omega_{\mathbf{W}}}$.

$$\vec{l_{\mathbf{W}}} = \vec{G} + \vec{h} + (\vec{\alpha_{\mathbf{W}}} \times \vec{D_{i\mathbf{W}}}) + (\vec{\omega_{\mathbf{W}}} \times (\vec{\omega_{\mathbf{W}}} \times \vec{D_{i\mathbf{W}}})) \tag{2.4}$$

ASSUMPTION: The IMU is only rotating in the gimbal, which is theoretically true if all the gimbal ring axes are aligned to be concentric. If the IMU is not translating, but just rotating in space, then $\mathbf{S}$ and $\mathbf{W}$ remain concentric, $\vec{h} = 0$, $|\vec{S_{i\mathbf{W}}}| = |\vec{S_{i\mathbf{S}}}|$,

$|\vec{V}_{i\mathbf{W}}| = |\vec{V}_{i\mathbf{S}}|$ and $R_{S\to W}$ represents the 3-by-3 rotation matrix which transforms vectors from **S** to **W**. Equation (2.2) can be expanded to

$$A_i = \vec{S}_{i\mathbf{W}} \circ (\vec{G} + (\vec{\alpha}_{\mathbf{W}} \times \vec{D}_{i\mathbf{W}}) + (\vec{\omega}_{\mathbf{W}} \times (\vec{\omega}_{\mathbf{W}} \times \vec{D}_{i\mathbf{W}}))) + \mu_i \tag{2.5}$$

Vectors in **S** can be rotated to **W** by multiplying by $R_{S\to W}$.

$$A_i = (R_{S\to W}\cdot\vec{S}_{i\mathbf{S}})\circ(\vec{G}+(\vec{\alpha}_{\mathbf{W}}\times(R_{S\to W}\cdot\vec{D}_{i\mathbf{S}}))+(\vec{\omega}_{\mathbf{W}}\times(\vec{\omega}_{\mathbf{W}}\times(R_{S\to W}\cdot\vec{D}_{i\mathbf{S}}))))+\mu_i \tag{2.6}$$

The dot product can be turned into a matrix transpose product.

$$A_i = (R_{S\to W}\cdot\vec{S}_{i\mathbf{S}})^T\cdot(\vec{G}+(\vec{\alpha}_{\mathbf{W}}\times(R_{S\to W}\cdot\vec{D}_{i\mathbf{S}}))+(\vec{\omega}_{\mathbf{W}}\times(\vec{\omega}_{\mathbf{W}}\times(R_{S\to W}\cdot\vec{D}_{i\mathbf{S}}))))+\mu_i \tag{2.7}$$

Because of the orthonormality of rotation matrices, the transpose of the rotation matrix is the same as the inverse of the rotation matrix. $R^T_{S\to W} = R_{W\to S}$

$$A_i = \vec{S}^T_{i\mathbf{S}}\cdot R_{W\to S}\cdot(\vec{G}+(\vec{\alpha}_{\mathbf{W}}\times(R_{S\to W}\cdot\vec{D}_{i\mathbf{S}}))+(\vec{\omega}_{\mathbf{W}}\times(\vec{\omega}_{\mathbf{W}}\times(R_{S\to W}\cdot\vec{D}_{i\mathbf{S}}))))+\mu_i \tag{2.8}$$

The rotational components of the linear acceleration can be represented in matrix form, and separated from $R_{S\to W}\cdot\vec{D}_{i\mathbf{S}}$. In particular, if $\vec{\alpha}_{\mathbf{W}} = (r,s,t)^T$, and $\vec{\omega}_{\mathbf{W}} = (u,v,w)^T$, then the matrix $f(\vec{\alpha}_{\mathbf{W}},\vec{\omega}_{\mathbf{W}})$ can be defined such that

$$(\vec{\alpha}_{\mathbf{W}}\times(R_{S\to W}\cdot\vec{D}_{i\mathbf{S}}))+(\vec{\omega}_{\mathbf{W}}\times(\vec{\omega}_{\mathbf{W}}\times(R_{S\to W}\cdot\vec{D}_{i\mathbf{S}}))) = f(\vec{\alpha}_{\mathbf{W}},\vec{\omega}_{\mathbf{W}})\cdot R_{S\to W}\cdot\vec{D}_{i\mathbf{S}} \tag{2.9}$$

$$f(\vec{\alpha}_{\mathbf{W}},\vec{\omega}_{\mathbf{W}}) = \begin{pmatrix} 0 & -t & s \\ t & 0 & -r \\ -s & r & 0 \end{pmatrix} + \begin{pmatrix} -v^2-w^2 & v\cdot u & w\cdot u \\ v\cdot u & -w^2-u^2 & w\cdot v \\ w\cdot u & w\cdot v & -u^2-v^2 \end{pmatrix} \tag{2.10}$$

The cross products in Equation (2.8) can be replaced by matrix multiplications.

$$A_i = \vec{S}^T_{i\mathbf{S}}\cdot R_{W\to S}\cdot(\vec{G}+f(\vec{\alpha}_{\mathbf{W}},\vec{\omega}_{\mathbf{W}})\cdot R_{S\to W}\cdot\vec{D}_{i\mathbf{S}})+\mu_i \tag{2.11}$$

So, the problem of calibrating an accelerometer comes down to estimating μ_i, $\vec{S}_{i\mathbf{S}}$, and $\vec{D_{i\mathbf{S}}}$ in Equation (2.11) given a set of discrete datapoints, $A_i[k]$, $R_{W\to S}[k]$, $\vec{\alpha_{\mathbf{W}}}[k]$, and $\vec{\omega_{\mathbf{W}}}[k]$.

2.2.2 Rate Gyro Equations

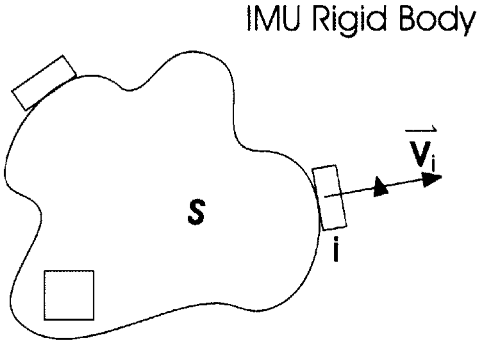

Figure 2-3: Rate Gyro i Situated on the IMU in **S**

Rate gyro i is situated in **S**, (Figure 2-3), with

- orientation vector $\vec{V_{i\mathbf{S}}} = (v_x, v_y, v_z)^T$
- scale factor $|\vec{V_{i\mathbf{S}}}|$
- offset voltage χ_i
- output voltage Y_i

Because the rate gyros measure *angular* velocity, they can be translated to an arbitrary position on the rigid body IMU and will still measure the same value. Thus, the position vector cannot be measured for the rate gyros, as it does not affect the output Y_i.

A rate gyro measures a component of the angular velocity of the IMU in the world reference frame, $\omega_{\mathbf{W}}$, and outputs a voltage equal to the projection of $\vec{\omega_{\mathbf{W}}}$ onto the rate gyro orientation vector $\vec{V_{i\mathbf{W}}}$, multiplied by some scaling, $|\vec{V_{i\mathbf{W}}}|$, plus some offset voltage χ_i.

$$Y_i = |\vec{V_{i\mathbf{W}}}| \cdot Proj_{\vec{V_{i\mathbf{W}}}}(\vec{\omega_{\mathbf{W}}}) + \chi_i \tag{2.12}$$

Equation (2.12) can be simplified to

$$Y_i = \vec{V_{i\mathbf{S}}}^T \cdot R_{S \to W} \cdot \vec{\omega_{\mathbf{W}}} + \chi_i \tag{2.13}$$

So, the problem of calibrating a rate gyro comes down to estimating χ_i and $\vec{V_{i\mathbf{S}}}$ in Equation (2.13) given a set of discrete datapoints, $Y_i[k]$, $R_{W \to S}[k]$, and $\vec{\omega_{\mathbf{W}}}[k]$.

2.3 Gimbal Equations

The gimbal is composed of 3 concentric rings, which allow the central ring to be spun in any orientation, as seen in (Figure 2-4). The gimbal has a rotation sensor at each of the axes of the rings, which measures the angle between rings. From this information, the rotation matrices and the angular velocity and acceleration vectors can be determined.

2.3.1 Determining $R_{S \to W}$

A right-hand coordinate system is chosen such that the inner ring spins about the y-axis of the middle ring by some θ_{in} as seen in (Figure 2-4). Recall that the IMU is rigidly fixed in the inner ring, such that the coordinate frame of the inner ring is **S**. Then, the rotation matrix that transforms a vector from the inner ring to the middle ring, $R_{S \to mid}$, is a y-axis rotation by θ_{in}.

$$R_{S \to mid}(\theta_{in}) = R_Y(\theta_{in}) \tag{2.14}$$

Similarly, the middle ring spins about the z-axis of the outer ring by some θ_{mid}.

$$R_{mid \to out} = R_Z(\theta_{mid}) \tag{2.15}$$

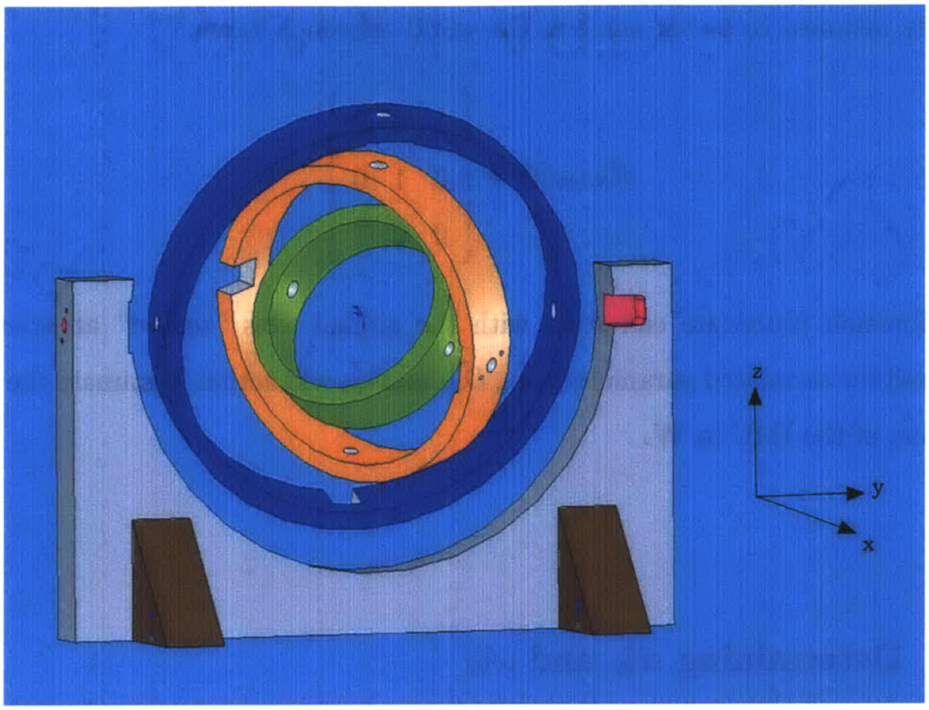

Figure 2-4: Solidworks Model of Gimbal, Right-Hand Coordinate System Defined

The outer ring spins about the y-axis of the world reference frame, **W** by some $\theta_{out.}$

$$R_{out \to W} = R_Y(\theta_{out}) \tag{2.16}$$

The total rotation matrix from **S** to **W** is the product of the intermediate rotations.

$$R_{S \to W} = R_{out \to W} \cdot R_{mid \to out} \cdot R_{S \to mid} \tag{2.17}$$

The rotation matrix $R_{S \to W}$ and its transpose, $R_{W \to S}$, can be represented in terms of gimbal measurements of rotations at each of the axes of the rings.

$$R_{S \to W} = R_Y(\theta_{out}) \cdot R_Z(\theta_{mid}) \cdot R_Y(\theta_{in}) \tag{2.18}$$

Note that the calibration algorithm does not constrain the initial state of how **S** is oriented in **W**. For the sake of simplicity, the orientation of the sensor reference frame is initially assumed to be the same as the world reference frame.

$$R_{S \to W}[0] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.19}$$

The calibration procedure can start with the gimbal axes "zeroed" at arbitrarily angles, and the estimated parameters $\vec{S_{i\mathbf{S}}}, \vec{D_{i\mathbf{S}}}, and\vec{V_{i\mathbf{S}}}$ will actually estimate the initial orientation of the IMU in $\mathbf{W}$.

2.3.2 Determining $\vec{\alpha_{\mathbf{W}}}$ and $\vec{\omega_{\mathbf{W}}}$

$\vec{\alpha_{\mathbf{W}}}$ represents the angular acceleration of the IMU in the world reference frame. The magnitude of $\vec{\alpha_{\mathbf{W}}}$ is equal to the magnitude of the angular acceleration in (rad/sec^2). The normalized vector of $\vec{\alpha_{\mathbf{W}}}$ represents the axis about which the angular acceleration is acting. The magnitude of the angular acceleration of the IMU about the middle ring is $\ddot{\theta_{in}}$, and this acceleration occurs about the y-axis, $(0, 1, 0)^T$.

$$\vec{\alpha_{\mathbf{mid}}} = \ddot{\theta_{in}} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \tag{2.20}$$

The angular acceleration of the IMU about the outer ring, $\vec{\alpha_{\mathbf{out}}}$, is equal to the angular acceleration of the middle ring about the z-axis of the outer ring, $\ddot{\theta_{mid}} \cdot (0, 0, 1)^T$, plus the angular acceleration of the inner ring about the middle ring, $\vec{\alpha_{\mathbf{mid}}}$ rotated into the outer ring coordinate frame by $R_{mid \to out}$.

$$\vec{\alpha_{\mathbf{out}}} = \ddot{\theta_{mid}} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + R_{mid \to out} \cdot \vec{\alpha_{\mathbf{mid}}} \tag{2.21}$$

The angular acceleration of the IMU about $\mathbf{W}$ can be calculated in a similar manner.

$$\vec{\alpha_{\mathbf{W}}} = \ddot{\theta}_{out} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + R_{out \to W} \cdot \vec{\alpha_{\mathbf{out}}} \tag{2.22}$$

Equations (2.20), (2.21), and (2.22) can be combined.

$$\vec{\alpha_{\mathbf{W}}} = \ddot{\theta}_{out} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + R_{out \to W} \cdot \ddot{\theta}_{mid} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + R_{out \to W} \cdot R_{mid \to out} \cdot \ddot{\theta}_{in} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \tag{2.23}$$

Equations (2.14), (2.15), and (2.16) can be substituted into Equation (2.23) to yield the total equation for the angular acceleration of the IMU in $\mathbf{W}$ in terms of θ's and $\ddot{\theta}$'s.

$$\vec{\alpha_{\mathbf{W}}} = \ddot{\theta}_{out} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + R_Y(\theta_{out}) \cdot \ddot{\theta}_{mid} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + R_Y(\theta_{out}) \cdot R_Z(\theta_{mid}) \cdot \ddot{\theta}_{in} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \tag{2.24}$$

The angular velocity of the IMU about $\mathbf{W}$ can be calculated in a similar way to the angular acceleration, yielding a total equation for the angular acceleration of the IMU in $\mathbf{W}$ in terms of θ's and $\dot{\theta}$'s.

$$\vec{\omega_{\mathbf{W}}} = \dot{\theta}_{out} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + R_Y(\theta_{out}) \cdot \dot{\theta}_{mid} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + R_Y(\theta_{out}) \cdot R_Z(\theta_{mid}) \cdot \dot{\theta}_{in} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \tag{2.25}$$

2.4 LMMSE Estimation of IMU Parameters

As the IMU is rotated in space, the gimbal rotation sensor and the accelerometer and rate gyro output voltages are recorded. Calibration involves using the discrete datapoints $\theta_{in}[k]$, $\theta_{mid}[k]$, $\theta_{out}[k]$, $A_i[k]$ and $Y_i[k]$ to estimate the unknown IMU parameters, μ_i, $\vec{S_{i\mathbf{S}}}$ and $\vec{D_{i\mathbf{S}}}$ for each accelerometer i, and χ_j and $\vec{V_{j\mathbf{S}}}$ for each rate gyro j. The $\dot{\theta}[k]$'s and $\ddot{\theta}[k]$'s are approximated from the $\theta[k]$'s and the sampletime, δt.

$$\dot{\theta}[k] = \frac{(\theta[k] - \theta[k-1])}{\delta t} \tag{2.26}$$

$$\ddot{\theta}[k] = \frac{(\dot{\theta}[k] - \dot{\theta}[k-1])}{\delta t} \tag{2.27}$$

Note that the calibration algorithm assumes the system is time-invariant.

2.4.1 Batch Estimator

Batch estimators use all the datapoints in a set to estimate unknown parameters. As more and more datapoints are added to the set, multiplications and inversions need to be calculated on matrices of increasing size. The batch estimator is useful in a calibration scheme where a lot of datapoints are taken, and then post-processed.

Accelerometer Estimator

Equation (2.11) can be rewritten with reference to a datapoint at time index k.

$$A_i[k] = \vec{S^T_{i\mathbf{S}}} \cdot R_{W\to S}[k] \cdot (\vec{G} + f(\vec{\alpha_{\mathbf{W}}}[k], \vec{\omega_{\mathbf{W}}}[k]) \cdot R_{S\to W}[k] \cdot \vec{D_{i\mathbf{S}}}) + \mu_i \tag{2.28}$$

The offset voltage, μ_i can be pulled into a vector with the unknown orientation.

$$A_i[k] = (\vec{S^T_{i\mathbf{S}}}, \mu_i) \cdot \begin{bmatrix} R_{W\to S}[k] \cdot (\vec{G} + f(\vec{\alpha_{\mathbf{W}}}[k], \vec{\omega_{\mathbf{W}}}[k]) \cdot R_{S\to W}[k] \cdot \vec{D_{i\mathbf{S}}}) \\ 1 \end{bmatrix} \tag{2.29}$$

Note that for a single datapoint, Equation (2.29) has a matrix structure as follows.

$$[1 \times 1] = [1 \times 4] \cdot [4 \times 1] \tag{2.30}$$

Equation (2.29) can be written to include multiple datapoints.

$$\underbrace{[A_i[1] \dots A_i[n]]}_{O} = (\vec{S}^{T}_{i\mathbf{S}}, \mu_i) \cdot \underbrace{\begin{bmatrix} R_{W \to S}[1] \cdot (\vec{G} + f(\alpha_{\vec{\mathbf{W}}}[1], \omega_{\vec{\mathbf{W}}}[1]) \cdot R_{S \to W}[1] \cdot \vec{D}_{i\mathbf{S}}) & \dots \\ 1 & \dots \end{bmatrix}}_{K} \tag{2.31}$$

For a set of n datapoints, Equation (2.31) has a matrix structure as follows.

$$[1 \times n] = [1 \times 4] \cdot [4 \times n] \tag{2.32}$$

Because the system is non-linear in $(\vec{S}^{T}_{i\mathbf{S}}, \mu_i)$ and $\vec{D}_{i\mathbf{S}}$, there is no simple analytic way to simultaneously solve for $(\vec{S}^{T}_{i\mathbf{S}}, \mu_i)$ and $\vec{D}_{i\mathbf{S}}$. Various optimization techniques such as gradient descent or Newton-Rhapson iteration could be applied to solve for $(\vec{S}^{T}_{i\mathbf{S}}, \mu_i)$ and $\vec{D}_{i\mathbf{S}}$ simultaneously. Instead, an iterative approach is chosen in which $\vec{D}_{i\mathbf{S}}$ is assumed to be 0 and $(\vec{S}^{T}_{i\mathbf{S}}, \mu_i)$ is solved for directly. Next, $(\vec{S}^{T}_{i\mathbf{S}}, \mu_i)$ is fed back into Equation (2.31), and $\vec{D}_{i\mathbf{S}}$ is solved for directly. The updated estimate of $\vec{D}_{i\mathbf{S}}$ is used to update the estimate of $(\vec{S}^{T}_{i\mathbf{S}}, \mu_i)$ and so on until a sufficient level of convergence (i.e. minimum error threshold) is achieved. Equation (2.31) can now be represented as $O = (\vec{S}^{T}_{i\mathbf{S}}, \mu_i) \cdot K$, where O and K are known values and $(\vec{S}^{T}_{i\mathbf{S}}, \mu_i)$ is a vector of unknowns in the first step of the iteration. The Linear Minimum Mean Squared Estimate (LMMSE) can be achieved for a linear system by projecting the set of datapoints onto a lower-dimensional space spanned by the unknown components to be estimated. In this manner, the LMMSE takes the form of a simple matrix inversion using the pseudoinverse.

$$O \cdot K^T = (\vec{S}^{T}_{i\mathbf{S}}, \mu_i) \cdot K \cdot K^T \tag{2.33}$$

Note that $K \cdot K^T$ is invertible only if n is greater than or equal to 4. This mathematical constraint is another manifestation of the fact that it takes at least 4 sets of datapoints to solve for the 4 unknowns in $(\vec{S}^{T}_{i\mathbf{S}}, \mu_i)$. The LMMSE estimate of $(\vec{S}^{T}_{i\mathbf{S}}, \mu_i)$ then becomes

$$(\widehat{\vec{S}_{i\mathbf{S}}}^{T}, \widehat{\mu_i}) = O \cdot K^T \cdot (K \cdot K^T)^{-1} \tag{2.34}$$

To solve for $\vec{D}_{i\mathbf{S}}$ in the second step of the iteration, Equation (2.28) is rewritten.

$$A_i[k] - \vec{S}_{i\mathbf{S}}^T \cdot R_{W \to S}[k] \cdot \vec{G} - \mu_i = (\vec{S}_{i\mathbf{S}}^T \cdot R_{W \to S}[k] \cdot f(\vec{\alpha}_{\mathbf{W}}[k], \vec{\omega}_{\mathbf{W}}[k]) \cdot R_{S \to W}[k]) \cdot \vec{D}_{i\mathbf{S}} \tag{2.35}$$

Note that for a single datapoint, Equation (2.35) has a matrix structure as follows.

$$[1 \times 1] = [1 \times 3] \cdot [3 \times 1] \tag{2.36}$$

Equation (2.29) can be written to include multiple datapoints.

$$\underbrace{\begin{bmatrix} A_i[1] - \vec{S}_{i\mathbf{S}}^T \cdot R_{W \to S}[1] \cdot \vec{G} - \mu_i \\ \vdots \\ A_i[n] - \vec{S}_{i\mathbf{S}}^T \cdot R_{W \to S}[n] \cdot \vec{G} - \mu_i \end{bmatrix}}_{LHS} = \underbrace{\begin{bmatrix} (\vec{S}_{i\mathbf{S}}^T \cdot R_{W \to S}[1] \cdot f(\vec{\alpha}_{\mathbf{W}}[1], \vec{\omega}_{\mathbf{W}}[1]) \cdot R_{S \to W}[1]) \\ \vdots \\ (\vec{S}_{i\mathbf{S}}^T \cdot R_{W \to S}[n] \cdot f(\vec{\alpha}_{\mathbf{W}}[n], \vec{\omega}_{\mathbf{W}}[n]) \cdot R_{S \to W}[n]) \end{bmatrix}}_{RHS} \cdot \vec{D}_{i\mathbf{S}} \tag{2.37}$$

For a set of n datapoints, Equation (2.37) has a matrix structure as follows.

$$[n \times 1] = [n \times 3] \cdot [3 \times 1] \tag{2.38}$$

Equation (2.37) can now be represented as $LHS = RHS \cdot \vec{D}_{i\mathbf{S}}$, where LHS and RHS are known values. The LMMSE again takes the form of a simple matrix inversion using the pseudoinverse.

$$RHS^T \cdot LHS = RHS^T \cdot RHS \cdot \vec{D}_{i\mathbf{S}} \tag{2.39}$$

Note that $RHS^T \cdot RHS$ is invertible only if n is greater than or equal to 3 because it takes at least 3 sets of datapoints to solve for the 3 unknowns in $\vec{D}_{i\mathbf{S}}$. The LMMSE estimate of $\vec{D}_{i\mathbf{S}}^{\,T}$ then becomes

$$\widehat{\vec{D}_{i\mathbf{S}}}^{T} = (RHS^T \cdot RHS)^{-1} \cdot RHS^T \cdot LHS \tag{2.40}$$

Rate Gyro Estimator

The equation for estimating the rate gyro parameters, $\vec{V_{i\mathbf{S}}}$ and χ_i, is very similar to the first step of the iteration in estimating the accelerometer parameters. Equation (2.13) can be rewritten for a particular datapoint at time index k.

$$Y_i[k] = (\vec{V_{i\mathbf{S}}^T}, \chi_i) \cdot \begin{bmatrix} R_{W \to S}[k] \cdot \vec{\omega_{\mathbf{W}}}[k] \\ 1 \end{bmatrix} \tag{2.41}$$

Equation (2.41) can be written to include multiple datapoints.

$$[Y_i[1] \ldots Y_i[n]] = (\vec{V_{i\mathbf{S}}^T}, \chi_i) \cdot \begin{bmatrix} R_{W \to S}[1] \cdot \vec{\omega_{\mathbf{W}}}[1] & \ldots & R_{W \to S}[n] \cdot \vec{\omega_{\mathbf{W}}}[n] \\ 1 & \ldots & 1 \end{bmatrix} \tag{2.42}$$

When written in the $O = (\vec{V_{i\mathbf{S}}^T}, \chi_i) \cdot K$ structure, Equation (2.42) can be solved using the pseudoinverse using at least 4 datapoints. The LMMSE estimate of $(\vec{V_{i\mathbf{S}}^T}, \chi_i)$ then becomes

$$(\widehat{\vec{V_{i\mathbf{S}}}}^T, \widehat{\chi_i}) = O \cdot K^T \cdot (K \cdot K^T)^{-1} \tag{2.43}$$

2.4.2 Recursive Estimator

It should be noted that as the number of datapoints increases, the complexity and time of using the batch estimator will also increase. A recursive estimator which carries a constant amount of state information from one timestep to the next can be implemented to calibrate the sensor values. The recursive estimator follows the Predictor-Corrector structure, in which the angular velocity and linear acceleration vectors are predicted using the current estimated sensor parameters. The error between the predicted and actual angular velocity, and linear acceleration, is used to correct the estimate of the sensor parameters, [11]. Such an online technique will be useful later for a real-time, interactive calibration scheme, as noted in Chapter 8.

Chapter 3

Yaw, Pitch, Roll Estimation

Once an IMU has been calibrated, the forward problem remains of estimating the orientation and position of the IMU in space, based on the sensor data. Because the IMU discussed in this project is fixed inside a gimbal, the assumption of pure rotation still holds when attempting to predict the orientation, $R_{W \to S}$. Because most mechanical engineers prefer to think in terms of YPR angles, as opposed to rotation matrices, these parameters will be estimated and analyzed.

3.1 Estimating Linear Acceleration

Given an IMU with calibrated sensors, each of the accelerometers measures a different linear acceleration based on its position on the IMU. This is the result of $\vec{D_i}_{\mathbf{S}}$ in Equation (2.11). ASSUMPTION: The basic assumption behind all estimates of orientation from accelerometers is that the sensors act as inclinometers; i.e. gravity is the only component of the linear acceleration vector. According to this gross assumption, the linear acceleration felt by all the accelerometers is in fact the same, and is just $\vec{G}$. Note that this assumption is strictly only true if the IMU is not rotating or accelerating in space. The Hybrid Estimation approach discussed later in this chapter deals with ameliorating the errors induced by this assumption. Equation (2.11) can be written to incorporate m accelerometers, assuming the same linear acceleration,

$\vec{l}_{\mathbf{S}}[k] = R_{W \rightarrow S}[k] \cdot \vec{l}_{\mathbf{W}}[k]$ for each sensor.

$$\underbrace{\begin{bmatrix} A_1[k] - \mu_1 \\ \vdots \\ A_m[k] - \mu_m \end{bmatrix}}_{A_{TOT}} = \underbrace{\begin{bmatrix} \vec{S}^T_{1\mathbf{S}} \\ \vdots \\ \vec{S}^T_{m\mathbf{S}} \end{bmatrix}}_{S_{TOT}} \cdot \vec{l}_{\mathbf{S}}[k] \tag{3.1}$$

For m accelerometers, Equation (3.1) has the following matrix structure.

$$[m \times 1] = [m \times 3] \cdot [3 \times 1] \tag{3.2}$$

Again, using the pseudoinverse, the linear acceleration of the IMU, as seen in the sensor reference frame, can be estimated. Note the mathematical constraint that there must be at least 3 accelerometers, measuring 3 independent axes, in order to take the pseudoinverse of S_{TOT}.

$$\hat{\vec{l}}_{\mathbf{S}}[k] = (S^T_{TOT} \cdot S_{TOT})^{-1} \cdot S_{TOT} \cdot A_{TOT} \tag{3.3}$$

3.2 Estimating Angular Velocity

Because the rate gyros measure angular velocity, all the rate gyros measure components of the same angular velocity vector in the sensor reference frame $\vec{\omega}_{\mathbf{S}}[k] = R_{W \rightarrow S}[k] \cdot \vec{\omega}_{\mathbf{W}}[k]$. (2.13) can be written to incorporate o rate gyros.

$$\underbrace{\begin{bmatrix} Y_1[k] - \chi_1 \\ \vdots \\ Y_o[k] - \chi_o \end{bmatrix}}_{Y_{TOT}} = \underbrace{\begin{bmatrix} \vec{V}^T_{1\mathbf{S}} \\ \vdots \\ \vec{V}^T_{o\mathbf{S}} \end{bmatrix}}_{V_{TOT}} \cdot \vec{\omega}_{\mathbf{S}}[k] \tag{3.4}$$

Again, the IMU must have at least 3 independently oriented rate gyros in order to estimate the angular velocity. For o rate gyros, the angular velocity of the IMU, as seen in the sensor reference frame can be estimated.

$$\widehat{\omega_{\mathrm{S}}}[k] = (V_{TOT}^T \cdot V_{TOT})^{-1} \cdot V_{TOT} \cdot Y_{TOT} \tag{3.5}$$

3.3 Kalman Filter Estimation

Most of the literature involving IMUs describes methods of estimating orientation of an object using a Kalman filter. The basic idea behind the Kalman filter is that there is a process, with some associated process noise, and a measurement of the process with some associated measurement noise. The Kalman filter follows the Predictor-Corrector structure of estimation. The rate gyro outputs are used to estimate the angular velocity. The angular velocity is used to create an incremental rotation matrix which updates the predicted orientation of the gravity vector from one timestep to the next. The accelerometer outputs are measured, and used to correct the estimation of the orientation gravity vector. In this manner, the process noise corresponds to the noise induced by the rate gyros in creating a incremental rotation matrix, whereas the measurement noise corresponds to the noise in the accelerometers when estimating the gravity vector.

3.3.1 Hybrid Estimation

Generally, accelerometers on an IMU are used as inclinometers. Assuming low dynamic linear acceleration, the accelerometer readings are interpreted as pure projections of gravity. In this manner, the state of the accelerometers at a particular instant in time can predict the orientation of an object except for the yaw axis (rotation about gravity). Rate gyros, on the other hand, are subject to accumulated error because they must be integrated to estimate final orientation from angular velocities over time. Rate gyros are beneficial during periods of high dynamic linear acceleration, because the accelerometer readings are no longer accurate predictors of the gravity vector. In order to switch between using both the accelerometer and rate gyro information during periods of low acceleration and just using the rate gyro data during periods of high acceleration, a threshold ϵ and a factor $\rho[k]$ are defined.

$$\rho[k] = \begin{cases} 1 & : \quad |\widehat{\vec{l}_S}[k] - \vec{G}| < \epsilon \\ 0 & : \quad |\widehat{\vec{l}_S}[k] - \vec{G}| > \epsilon \end{cases} \tag{3.6}$$

3.3.2 Kalman Filter Implementation

ASSUMPTIONS: The noise between sensors is assumed to be uncorrelated (e.g. so that the noise signal on accelerometer i is independent of the noise signal on accelerometer j). Also, the noise profile itself is assumed to be zero-mean and Gaussian, which, as will be seen later in Chapter 5, will prove to be roughly true. These assumptions simplify the Kalman filter update eqations, [23]. The matrix $S(\vec{\omega}[k])$ is defined using components from an angular velocity vector $\vec{\omega}[k] = (\omega_x[k], \omega_y[k], \omega_z[k])^T$.

$$S(\vec{\omega}[k]) = \begin{bmatrix} 0 & \omega_z[k] & -\omega_y[k] \\ -\omega_z[k] & 0 & \omega_x[k] \\ \omega_y[k] & -\omega_x[k] & 0 \end{bmatrix} \tag{3.7}$$

The incremental rotation matrix, $R_{inc}[k]$, transforms a vector through 1 timestep δt based on the estimated angular velocity vector $\widehat{\vec{\omega}_S}[k]$ and $S(\widehat{\vec{\omega}_S}[k])$.

$$R_{W\to S}[k+1] = R_{inc}[k] \cdot R_{W\to S}[k] \tag{3.8}$$

$$R_{inc}[k] = I - \frac{S(\widehat{\vec{\omega}_S}[k])}{|\widehat{\vec{\omega}_S}[k]|}\sin(\widehat{\vec{\omega}_S}[k]\delta t) + \frac{S(\widehat{\vec{\omega}_S}[k])^2}{|\widehat{\vec{\omega}_S}[k]|^2}(1 - \cos(\widehat{\vec{\omega}_S}[k]\delta t)) \tag{3.9}$$

Assuming the rate gyros exhibit zero-mean Gaussian noise that is independent between sensors (i.e. zero covariance), the process noise can be modeled as q, the variance of the noise in $\widehat{\vec{\omega}_S}[k]$. Similarly, assuming the accelerometers exhibit zero-mean, independent Gaussian noise, the measurement noise can be modeled as r, the variance of the noise in $\widehat{\vec{l}_S}[k]$.

Kalman function update equations can be written to estimate $\widehat{\vec{g}_S}[k]$, the estimated gravity vector in the sensor reference frame.

$$p[k+1] = p[k] + q - \rho[k]\frac{p[k]^2}{p[k]+r} \tag{3.10}$$

$$\widehat{\vec{g}_{\mathbf{S}}}[k+1] = R_{inc}[k] \cdot \left(\widehat{\vec{g}_{\mathbf{S}}}[k] + \rho(k)\frac{p[k]}{p[k]+r}(\hat{\vec{l}}_{\mathbf{S}}[k] - \widehat{\vec{g}_{\mathbf{S}}}[k])\right) \tag{3.11}$$

$p[k]$ represents the updated noise correction term, and can be initialized to $p[0] = 0$. Because of the assumption that **S** and **W** are initially the same orientation in Equation (2.19), $\widehat{\vec{g}_{\mathbf{S}}}[0] = \vec{G}$. The Kalman estimate of $\widehat{\vec{g}_{\mathbf{S}}}[k]$ can be used to estimate the third column of the rotation matrix which transforms the gravity vector from the world reference frame to the sensor reference frame.

$$\vec{g}_{\mathbf{S}}[k] = R_{W\to S}[k] \cdot \vec{G} = \begin{bmatrix} r_{11}[k] & r_{12}[k] & r_{13}[k] \\ r_{21}[k] & r_{22}[k] & r_{23}[k] \\ r_{31}[k] & r_{32}[k] & r_{33}[k] \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ -9.81 \end{bmatrix} = -9.81 \cdot \begin{bmatrix} r_{13}[k] \\ r_{23}[k] \\ r_{33}[k] \end{bmatrix} \tag{3.12}$$

$R_{W\to S}$ can be represented in terms of yaw(α)-pitch(β)-roll(γ) rotations:

$$\begin{bmatrix} \cos(\alpha)\cos(\beta) & \sin(\alpha)\cos(\beta) & -\sin(\beta) \\ -\sin(\alpha)\cos(\beta) + \cos(\alpha)\sin(\beta)\sin(\gamma) & \cos(\alpha)\cos(\gamma) + \sin(\alpha)\sin(\beta)\sin(\gamma) & \cos(\beta)\sin(\gamma) \\ \sin(\alpha)\sin(\gamma) + \cos(\alpha)\sin(\beta)\cos(\gamma) & \sin(\alpha)\sin(\beta)\cos(\gamma) - \cos(\alpha)\sin(\gamma) & \cos(\beta)\cos(\gamma) \end{bmatrix} \tag{3.13}$$

Note that the third column of $R_{W\to S}$ only contains β and γ terms.

$$\begin{bmatrix} r_{13}[k] \\ r_{23}[k] \\ r_{33}[k] \end{bmatrix} = \begin{bmatrix} -\sin(\beta[k]) \\ \cos(\beta[k])\sin(\gamma[k]) \\ \cos(\beta[k])\cos(\gamma[k]) \end{bmatrix} \tag{3.14}$$

The pitch term can be estimated from the first term in Equation (3.14). Using the estimated pitch term, the roll term can be estimated using second term in Equation (3.14). While β and γ are back-solved using the Kalman equations, α must be solved for in another way. Intuitively, accelerometers, when used to measure projections of gravity, do not and cannot measure any yaw component (i.e. a component perpendicular to the gravity vector). There may be a way to re-estimate the angular velocity vector from the Kalman estimated gravity vector, and use this re-estimated angular velocity vector to estimate the yaw angle. Some thoughts on this method are described in Chapter 8.

Current techniques for estimating yaw utilize only the rate gyro sensor data. Note that while the initial yaw component can be arbitrarily assigned, it is assumed to be 0 degrees. The vector $\vec{M}$ is defined as a static reference vector in the world coordinate frame, $(0,1,0)^T$. The rotated counter-part of $\vec{M}$ in the sensor reference frame is $\vec{m}_{\mathbf{S}}$.

$$\vec{m}_{\mathbf{S}}[k] = R_{W\to S}[k] \cdot \vec{M} = \begin{bmatrix} r_{11}[k] & r_{12}[k] & r_{13}[k] \\ r_{21}[k] & r_{22}[k] & r_{23}[k] \\ r_{31}[k] & r_{32}[k] & r_{33}[k] \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} r_{12}[k] \\ r_{22}[k] \\ r_{32}[k] \end{bmatrix} \tag{3.15}$$

Again, because **S** and **W** are initially the same orientation, $\widehat{\vec{m}_{\mathbf{S}}}[0] = (0,1,0)^T$. An update equation can be defined for $\widehat{\vec{m}_{\mathbf{S}}}[k]$ based solely on the information provided by the rate gyro sensors.

$$\widehat{\vec{m}_{\mathbf{S}}}[k+1] = R_{inc}[k] \cdot \widehat{\vec{m}_{\mathbf{S}}}[k] \tag{3.16}$$

Note that $\widehat{\vec{m}_{\mathbf{S}}}[k]$ will be subject to increasing accumulated errors, because there is no corrector term to retrue the prediction after each timestep. The vectors, $\vec{M}$ and $\vec{m}_{\mathbf{S}}$ are defined so as to extract the second column of the rotation matrix.

$$\begin{bmatrix} r_{12}[k] \\ r_{22}[k] \\ r_{32}[k] \end{bmatrix} = \begin{bmatrix} \sin(\alpha)\cos(\beta) \\ \cos(\alpha)\cos(\gamma) + \sin(\alpha)\sin(\beta)\sin(\gamma) \\ \sin(\alpha)\sin(\beta)\cos(\gamma) - \cos(\alpha)\sin(\gamma) \end{bmatrix} \tag{3.17}$$

Now, the yaw angle, α, can be estimated from any one of the terms in Equation (3.17), using the estimated roll and pitch angles form Equation (3.14).

Chapter 4

Hardware

The physical hardware used to calibrate the IMU includes the IMU, the gimbal, and data acquisition hardware, (Figure 4-1).

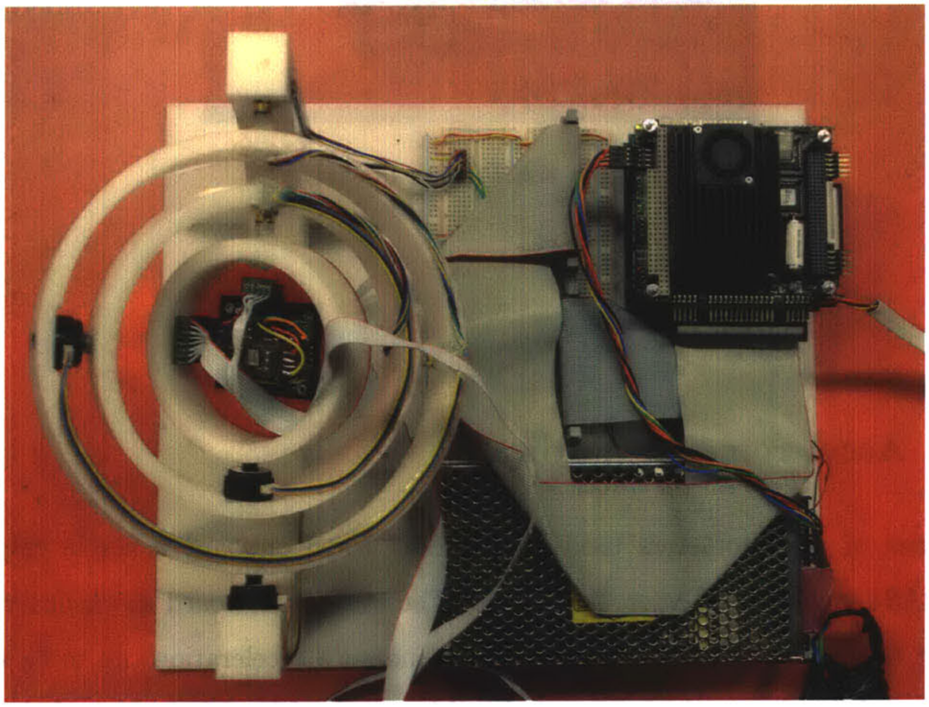

Figure 4-1: Picture of Complete Hardware Assembly

4.1 IMU Hardware

The IMU used in this thesis is composed of 3-axes of rate gyros and 3-axes of accelerometers. Both the sensors output analog voltages with an associated sensitivity and noise. Below is a picture of the completed IMU, (Figure 4-2). The rectangular green board is an evaluation board which contains the accelerometer hardware. The tubular looking structure made out of yellow fiberglass boards contains the rate gyro hardware.

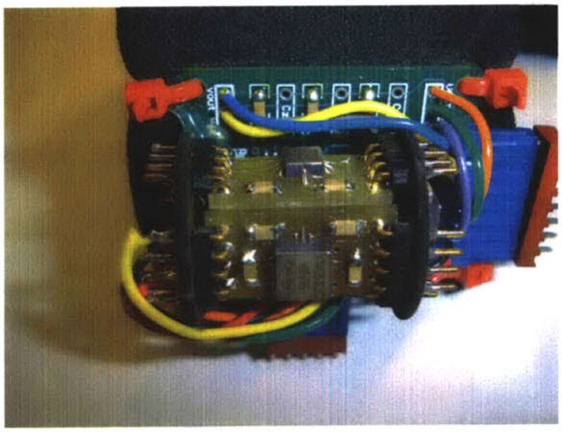

Figure 4-2: Picture of IMU

4.1.1 Accelerometer Hardware

The 3-axes of accelerometers used in the IMU come onboard a single chip, the LIS3L02AS4, made by STMicroelectronics. The evaluation board downconverts the 5V input to run the chip off of 3.3V, and can measure accelerations up to 6g's. The output signal for the accelerometers is roughly around 333mV per 1g and the zero-acceleration (offset voltage) for each axis is roughly 1.6V. The accelerometer axes have a misalignment of up to 4 degrees, [14]. As of August 2005, the cost of each of these 3-axis accelerometer chips is $20.

4.1.2 Rate Gyro Hardware

There are 3 separate Analog Devices ADXRS300 rate gyros. Each gyro runs off of 5V, and can measure up to 300 degrees of rotation per second. The sensitivity of the output signal for the gyros is typically 5mv per 1 deg/sec per V and the zero-voltage (offset voltage) is typically around 2.5V, [10]. As of August 2005, the cost of each of these rate gyro chips is $30. Note that the rate gyros are placed roughly orthogonal to one another via. design choice, though nowhere in the calibration algorithm is orthogonality of sensors specified or assumed.

4.2 Gimbal Hardware

The gimbal, designed in Solidworks, consists of 3 concentric rings and a base, (Figure 4-3). Each of the rings is connected to an outer piece along one axis. The gimbal is machined out of Delrin, as opposed to steel or aluminum, to reduce the potential of electromagnetic noise. One end of the connecting axis is created out of a male/female pair of 6-conductor audio stereo jack connections, made by CUI Inc. The other end of the connecting axis is created out of an aluminum dowel pin shaft, on which end a US Digital E4P press-fit-shaft rotary encoder is attached. The rotary encoders count 300 pulses per revolution (PPR) using quadrature encoding, so there are really 1200

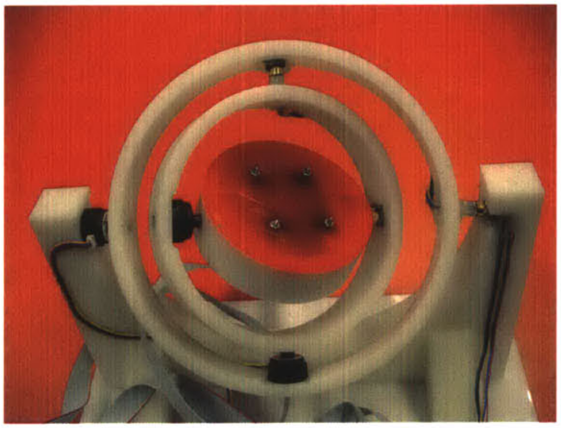

Figure 4-3: Picture of Gimbal

discrete steps which can be measured along a 360 degree revolution, leading to a resolution of 0.3 degrees.

4.3 Data Acquisition Hardware

All of the data acquisition hardware, as well as the gimbal and IMU hardware is run off of a 5 volt regulated power supply. A PC104 stack is used to record measurements from the gimbal and the IMU. The processor card for the PC104 stack is a Digital Logic MSMP3SEV board with an Intel Pentium III 700Mhz processor onboard. The A/D card used is a Diamond Systems MM32-AT board. It is used to record the 6 analog IMU signals. The quadrature encoder card used is a Sensoray 526 board. It is used to record the 3 rotary encoder signals from the gimbal. The PC104 processor card is running the Mathworks Matlab kernel, and the software to record the sensor outputs is written in Matlab and dowloaded to the PC104 using XPC. All the sensors are sampled at 1 Khz.

Chapter 5

Noise Estimation

The accelerometers and rate gyroscopes are subject to electrical and mechanical noise. Potential electrical noise sources include the "regulated" power supply, the PC104 stack, the analog to digital conversion, the board layout of the IMU, and switching induced by the quadrature encoders. Mechanical noise elements include human factors such as slight hand tremors when manipulating the IMU in the gimbal, and environmental factors such as gimbal axes misalignment.

5.1 Accelerometer Noise

In order to quantify the electrical noise of the accelerometers, 10 trials of experiments are run in which the IMU is left in a particular orientation in the gimbal. The trials are run for 60 seconds, which at the 1Khz sample rate corresponds to 60,000 datapoints per trial. A sample plot of an accelerometer output over the course of a minute shows a noise range of roughly 0.05V, which corresponds to roughly .05g, (Figure 5-1).

The means of the accelerometer signals change from trial to trial based on the IMU orientation. The noise profile, however, consistently looks like a Gaussian curve with a longer right tail and a smaller bimodal hump for all accelerometers, (Figure 5-2). For a specific accelerometer, the variance of the noise stays the same between trials.

While the histogram is a useful qualitative tool for determining the shape of the

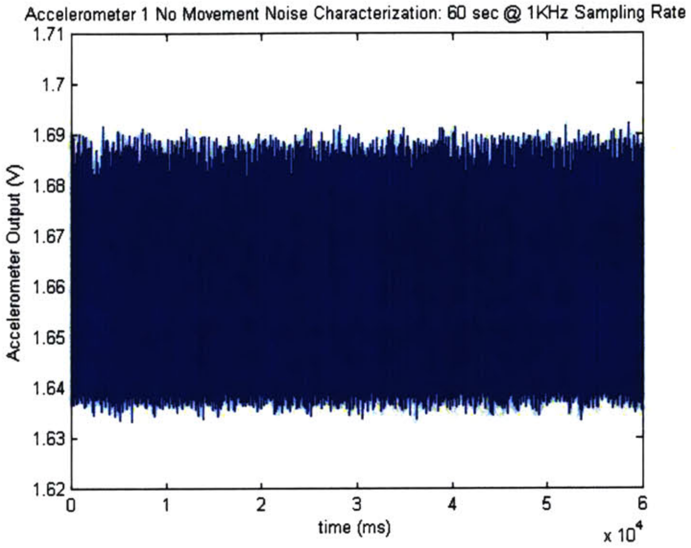

Figure 5-1: Accelerometer Noise Trial (60 sec)

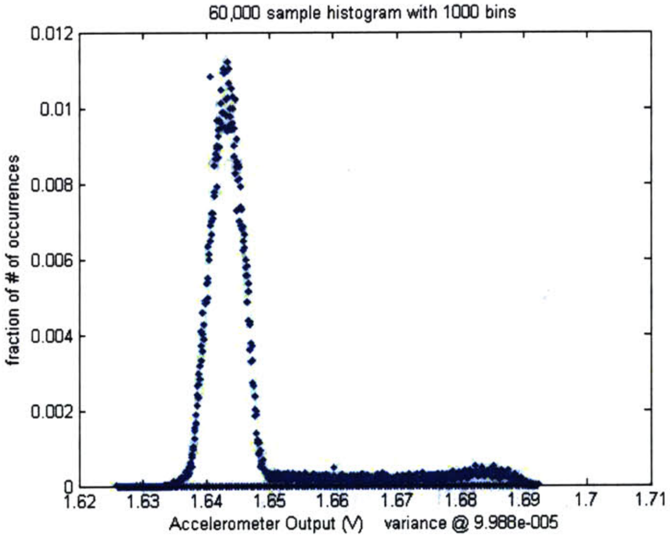

Figure 5-2: Accelerometer Histogram

electrical noise, it does not lend itself to comparisons between trials with varying sample sizes and bin sizes. Instead, a cumulative distribution function (CDF) can be generated to compare the noise profile between trials. The CDF is a probabilistic tool for representing a distribution and is effectively the integral of the PDF for a signal with analog distribution. By centering the means of all the trials at zero, the noise profile of the accelerometer can be seen to take on the same shape for all trials, (Figure 5-3).

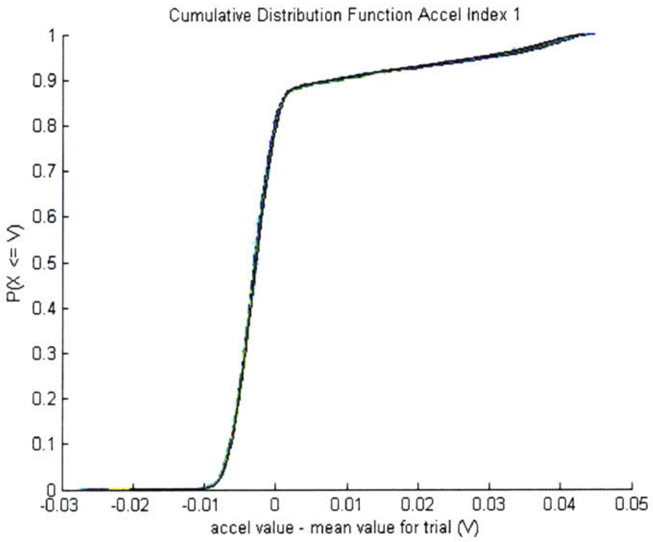

Figure 5-3: Accelerometer CDFs (10 Trials)

The accelerometer noise is modeled as a zero-mean Gaussian function to fit the noise model required by the Kalman filter, as described in Chapter 3. The power spectral density (PSD) for Additive White Gaussian Noise (AWGN) is a flat line in the frequency domain. The PSD of the accelerometer output has a peak at 0Hz corresponding to the DC value, and several higher-frequency peaks above 70Hz, (Figure 5-4). The region between 0Hz and 70Hz looks roughly like AWGN, and will be used to design a noise filter in Chapter 6.

5.1.1 Estimating the Kalman Measurement Noise, r

As described in Chapter 3, the measurement noise of the Kalman filter corresponds to the noise of the prediction of the gravity vector from the accelerometers. The simplified Kalman estimation function assumes that the measurement noise is AWGN, and that the covariance between accelerometers is zero. These assumptions are justified by the roughly Gaussian shape of the accelerometer noise profile, and by the covariance matrix of the accelerometer signals, (Table 5.1). In particular, the noise covariance

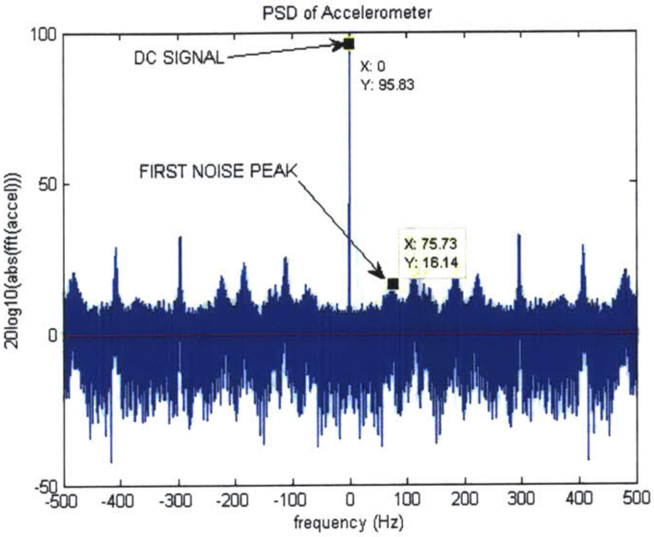

Figure 5-4: Accelerometer Signal PSD

$$noisecovariance(A_1, A_2, A_3) = \begin{pmatrix} 0.3343 & 0.0011 & -0.0126 \\ 0.0011 & 0.3318 & 0.0027 \\ -0.0126 & 0.0027 & 0.3329 \end{pmatrix} \cdot 10^{-4}$$

Table 5.1: Accelerometer Covariance Matrix

between accelerometers is minimal (at worst 1/30th) in comparison to the variance of the individual accelerometers (the diagonal elements). Also note that all the accelerometers have roughly the same noise variance. For this reason, the accelerometer noise variance is chosen as 3.3e-5V, and will be divided by the accelerometer scaling factor to determine r.

5.2 Rate Gyro Noise

The rate gyro signals are recorded during the same 10 trials as the accelerometers. A sample plot of a rate gyro output over the course of a minute shows a noise range of roughly 0.035V, which corresponds to roughly 7 deg/sec, (Figure 5-5). The means of the gyro signals during no movement are independent of the IMU orientation, and correspond to the offset voltage (i.e. the rate gyro output voltage at 0 deg/sec). The

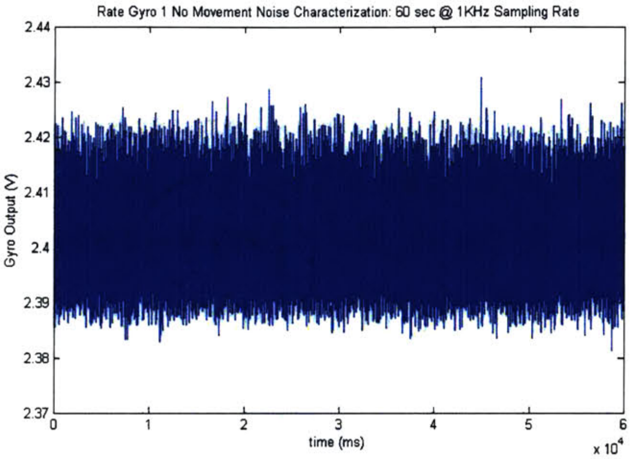

Figure 5-5: 60 sec of rate gyro noise trial

noise profile also looks like a Gaussian curve with a longer right tail and a smaller, yet closer, bimodal hump for all rate gyros, (Figure 5-6). For a specific rate gyro, the variance of the noise stays the same between trials.

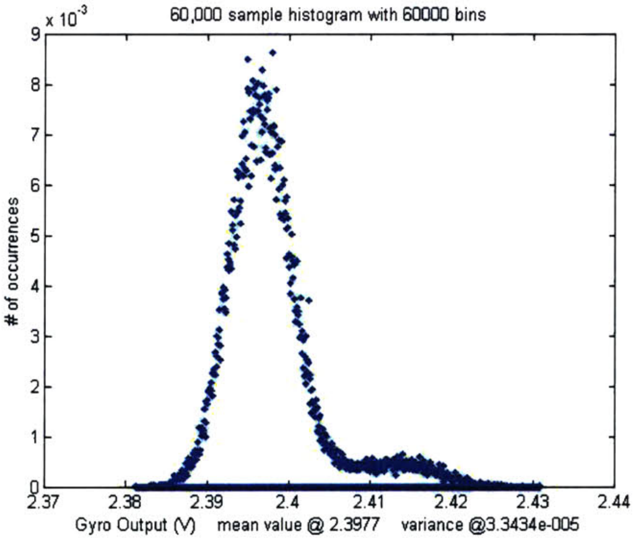

Figure 5-6: Rate Gyro Histogram

The CDFs of the rate gyroscopes are not zero-meaned, because they ought to have the same offset voltage from trial to trial. The noise profile of the rate gyros can be seen to take on the same, yet slightly mean shifted, shape for all trials, (Figure

5-7). The mean shifting corresponds to fluctuations in the rate gyro offset voltage. Moving means for gyros are attributed to temperature drift and electrical noise for rate gyros, and can lead to a large accumulated error when integrated over time to determine orientation. Over the course of a few hours, the offset voltages shift by a range of 3mV, or roughly 0.6 deg/sec. The current linear model does not account for the variation in rate gyro offset voltages.

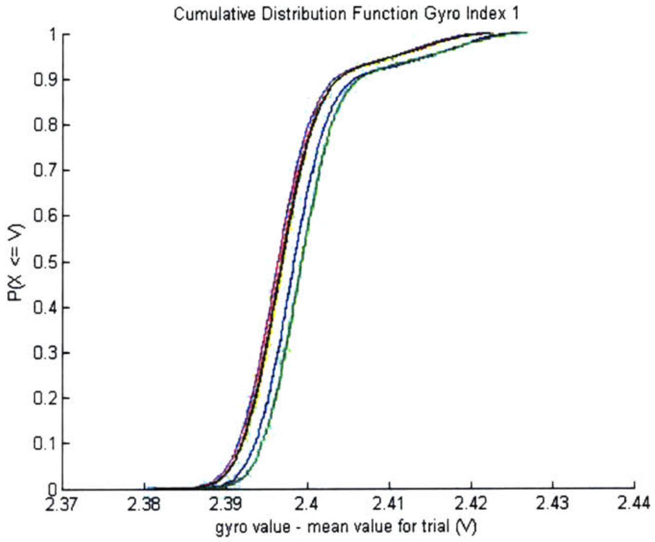

Figure 5-7: Rate Gyro CDFs (10 Trials)

The rate gyro noise is modeled as AWGN to fit the process model required by the Kalman filter. Again, the PSD of the rate gyro output has a peak at 0Hz corresponding to the DC value, and several higher-frequency peaks above 70Hz, (Figure 5-8). Again, the region between 0Hz and 70Hz looks roughly like AWGN, and will be used to design a noise filter in Chapter 6.

5.2.1 Estimating the Kalman Process Noise, q

The process noise of the Kalman filter corresponds to the noise of updating the estimate of the gravity vector due to the rate gyros. Again, the simplified Kalman estimation function assumes that the process noise is AWGN, and that the covariance between rate gyros is zero. These assumptions are justified by the roughly Gaussian

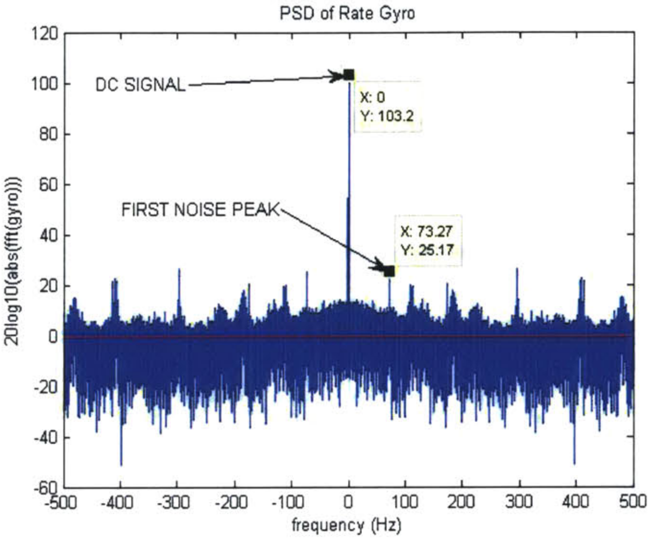

Figure 5-8: Rate Gyro Signal PSD

$$noisecovariance(Y_1, Y_2, Y_3) = \begin{pmatrix} 0.9988 & -0.0781 & -0.0811 \\ -0.0781 & 0.9949 & -0.0806 \\ -0.0811 & -0.0806 & 0.9964 \end{pmatrix} \cdot 10^{-4}$$

Table 5.2: Rate Gyro Covariance Matrix

shape of the rate gyro noise profile, and by the covariance matrix of the rate gyro signals, (Table 5.2). In particular, the noise covariance between rate gyros is minimal (at worst 1/12th) in comparison to the variance of the individual rate gyros (the diagonal elements). Also note that all the rate gyros have roughly the same noise variance. For this reason, the rate gyro noise variance is chosen as 1.0e-4V, and will be divided by the rate gyro scaling factor to determine q.

5.3 Gimbal Discretization

While the IMU sensors output analog voltages with associated noise, the rotary encoders on the gimbal output digital quadrature encoder bits, which have no associated analog noise. The gimbal measurements are, however, subject to discretization, in which the continuous angular rotation of the gimbal is measured in discrete steps

(Figure 5-9). The effects of discretization are amplified when higher order derivatives of the angle measurements are taken. A filter to smooth out these "choppy" signals to something more continuous is described in Chapter 6.

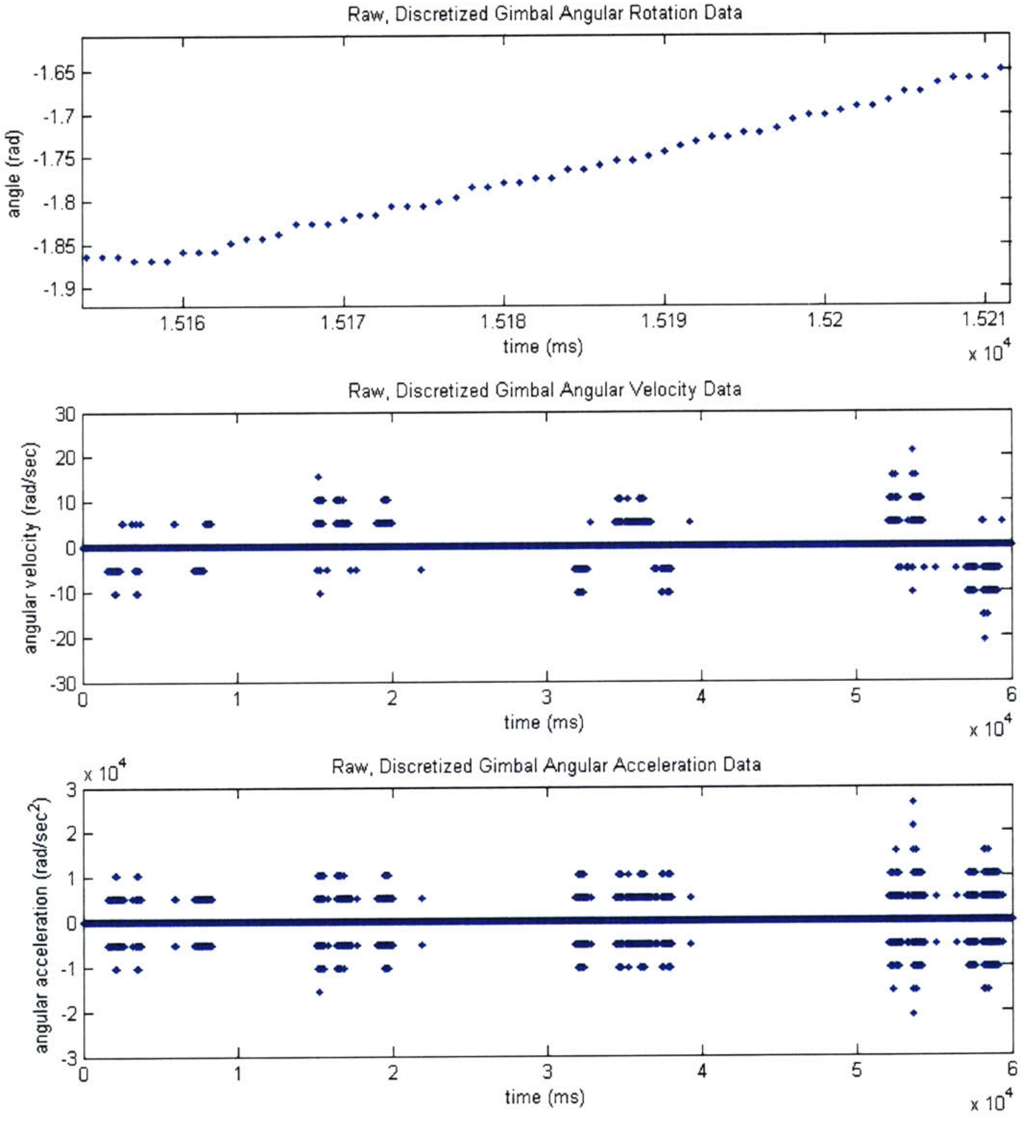

Figure 5-9: Effects of Gimbal Discretization

Chapter 6

Filter Determination

Filters must be designed to compensate for the electrical noise in the IMU and the discretization in the rotary encoders. In general, the filter used will be a low-pass filter to average out noise or to smooth a discrete signal, and its higher order derivatives, into a continuous one.

6.1 Zero-Phase-Lag Filters

The calibration process described so far is a batch process; i.e. all the data is first collected, and then the parameters are estimated. Therefore, the set of available filters can include non-causal filters, which can "look ahead" and use future data to compute a present value. The *filtfilt* command in Matlab is used to implement the low-pass filter. It ensures zero-phase-lag by running the data through the filter once, and then running the filtered data through the same filter, but backwards. Given a filter with a particular magnitude and phase plot, filtfilt generates an overall filter with twice the bode magnitude, and 0 phase for all frequencies.

6.1.1 IMU Low-Pass-Filters

Recall that both the accelerometer and rate gyro PSDs, (Figures 5-4 and 5-8), express noise peaks above 70Hz. The low-pass filters need to be designed to remove noise higher than 70Hz, yet still maintain a reasonable bandwidth. In the current design, the user must manually rotate the rings of the gimbal, which can happen with a frequency of at most 5Hz. So, the low-pass cutoff frequency must lie between 5Hz and 70Hz.

The filters to consider include the Moving Average Window (MAW) filter, and the Butterworth filter. As described in Chapter 8, the calibration algorithm will eventually be implemented on a stand-alone microcontroller or FPGA. Toward that end, the Butterworth filter is easier to implement using analog circuitry, whereas the MAW filter is easier to implement using digital operations. While both filters have qualitatively the same effect on the data, the MAW filter is chosen for ease of computation and increased performance.

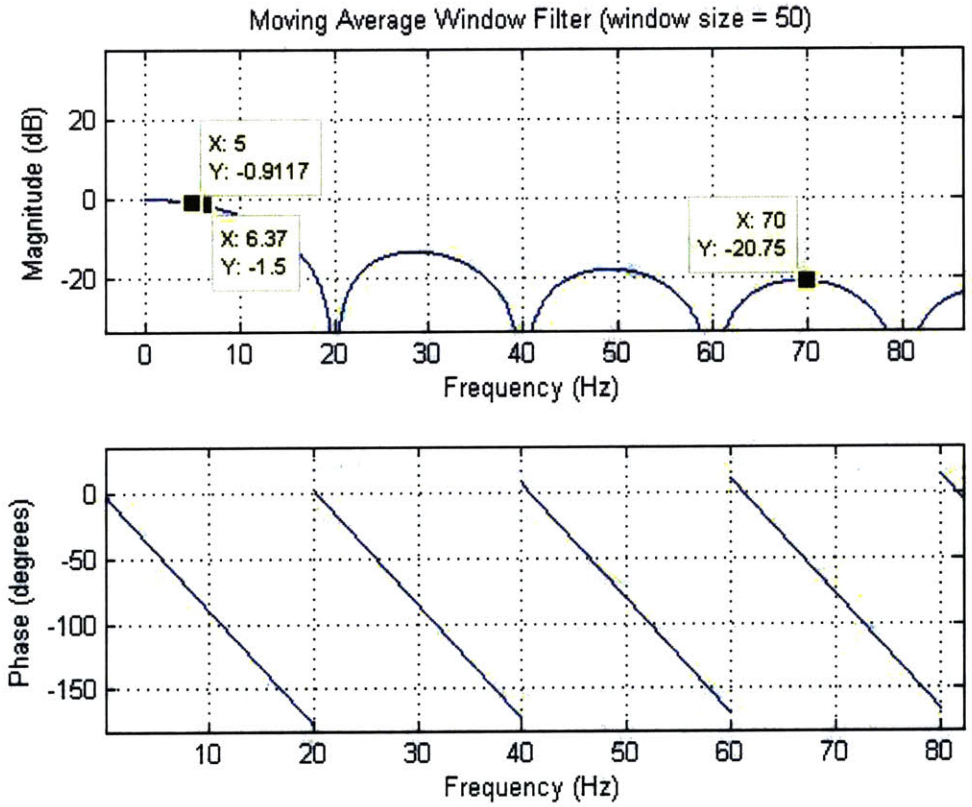

Figure 6-1: Bode Magnitude, Phase Plot of MAW Filter

The MAW window size of 50 is chosen to produce a low-pass filter with a -1.5dB point at 6.37Hz, and thus, an overall filter with a -3dB point at 6.37Hz when used with *filtfilt*, (Figure 6-1). The overall filter has a near-unity magnitude of more than .8, (-1.824dB), at 5Hz, and a minimizing magnitude of 0.01, (-41.5dB), at 70Hz. Qualitatively, a window size of 50 ensures that the noise is smoothed in a continuous, monotonic fashion while still preserving important features in the data, (Figure 6-2).

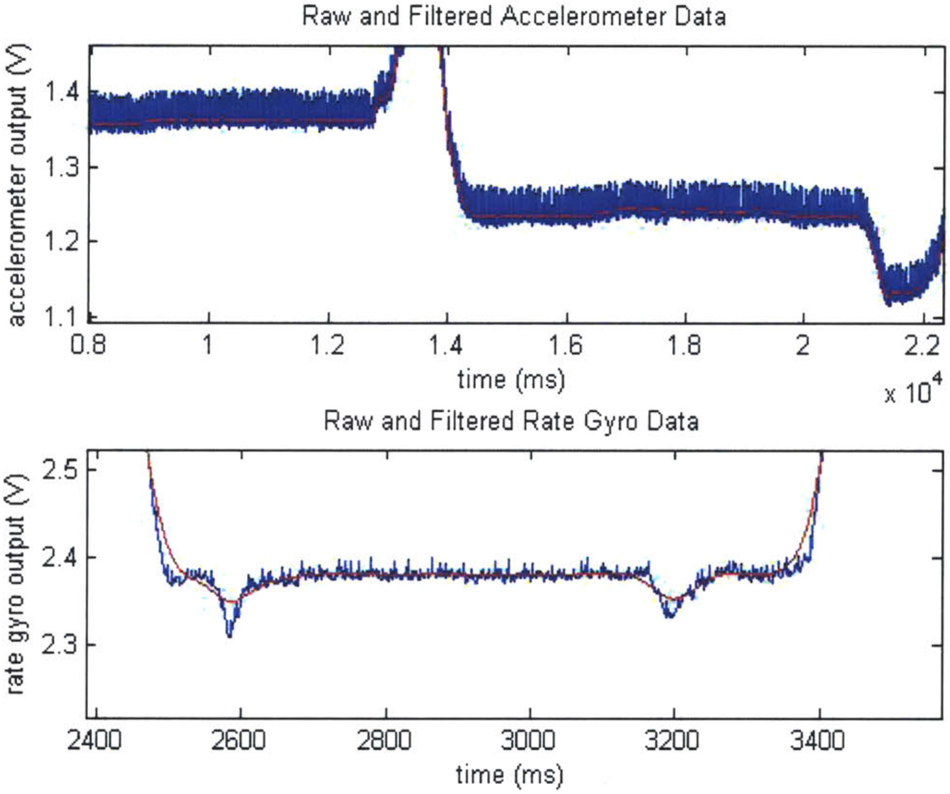

Figure 6-2: Effects of Filtering Noisy IMU Data

6.1.2 Rotary Encoder Smoothing Filters

The filter size and type of filter used for smoothing out the discretized rotary encoder signals is chosen to be the same as the one used for canceling the noise in the IMU, because the encoders are subject to the same miminum 5Hz bandwidth as the IMU sensors. Using the same filter eases the complexity of the computations as well.

Qualitatively, the angular velocities and angular accelerations look smooth, and have reasonable values of 5 rad/sec and 50 rad/sec^2, (Figure 6-3).

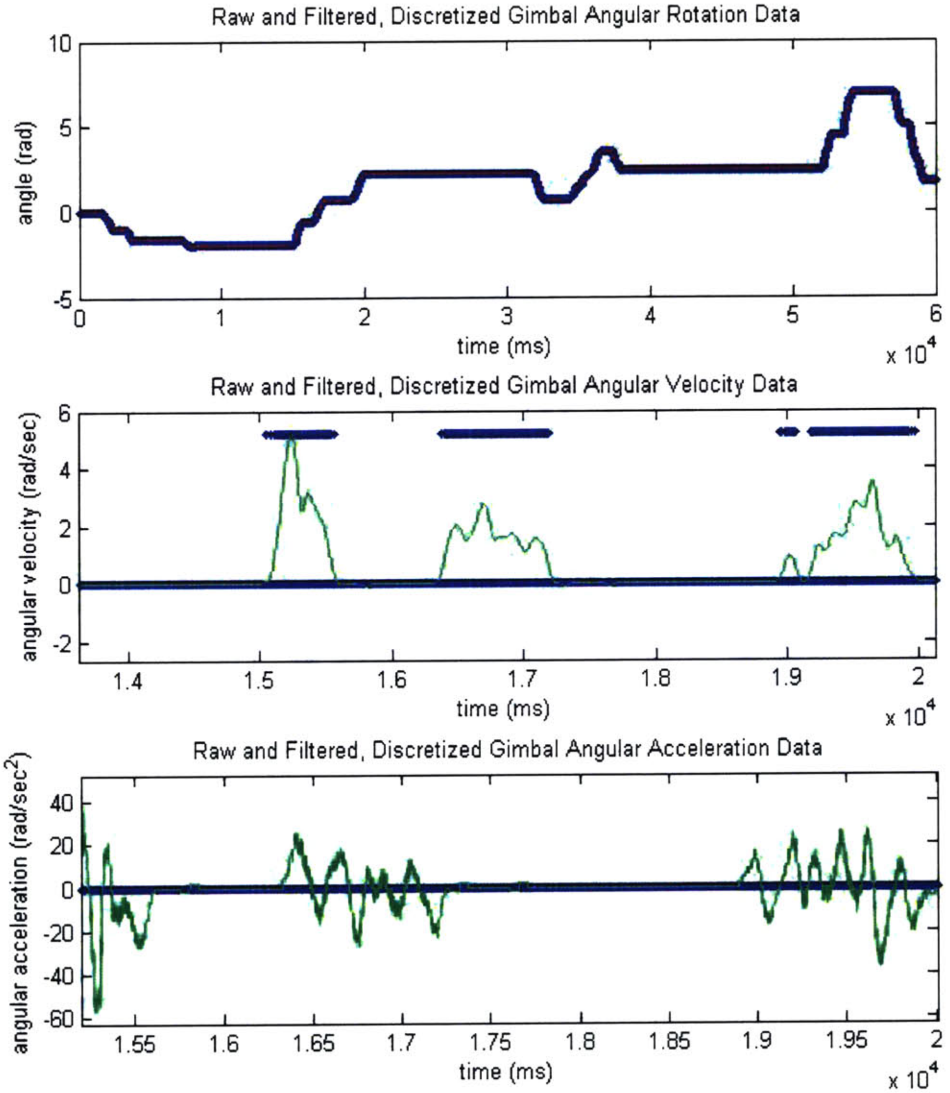

Figure 6-3: Effects of Filtering Discretized Gimbal Data

6.2 Real-Time Filters

When signals need to be processed in real-time, the filters used must be causal-filters, which only use data from the past. The inherent problem with real-time filters is that they induce a definite phase lag in the output signal from the input signal. In the case of using the MAW filter on its own (i.e. without using *filtfilt*), the phase-lag is approximately 25ms (roughly half the window size). Alternatively, the MAW filter can be used with *filtfilt* as described above, with a known input to output delay (i.e phase delay) of at least 50ms.

Chapter 7

Results

The IMU is spun inside the gimbal for 5 minutes and the IMU and gimbal sensors are sampled at 1Khz. 300,000 data points are recorded. The accelerometer, rate gyro, and rotary encoder signals are zero-phase-lag, low-pass filtered using a MAW filter with a window length of 50. The filtered rotary encoder signals are linearly converted into angles, differentiated to determine the angular velocity about each of the gimbal axes, and differentiated again to obtain the angular acceleration about the gimbal axes, (Figure 7-1).

Note that the amount of time and number of data points used is not enough to specify a "good" calibration routine. For example, the orientation could be left the same for the entire period of data collection. Recall from Chapter 2 that in order to solve the calibration equations, at least 4 unique data points need to be evaluated. While the model used to map the sensor parameters to the sensor outputs is a linear model, the system itself may contain some nonlinearities. When rotating the IMU inside the gimbal, care is taken to sweep through as many orientations as possible, in order to grab a rich enough data set. Ideally, the data set would contain an equal number of data points from a equally distributed range of angular velocity and angular acceleration vectors. The notion of quantifying the spatial representation of a "good" input is covered in the Chapter 8. The first 3 minutes of data (180,000 data points) are used to calibrate the IMU parameters. The later 2 minutes of data (120,000 data points) are used when estimating the YPR values.

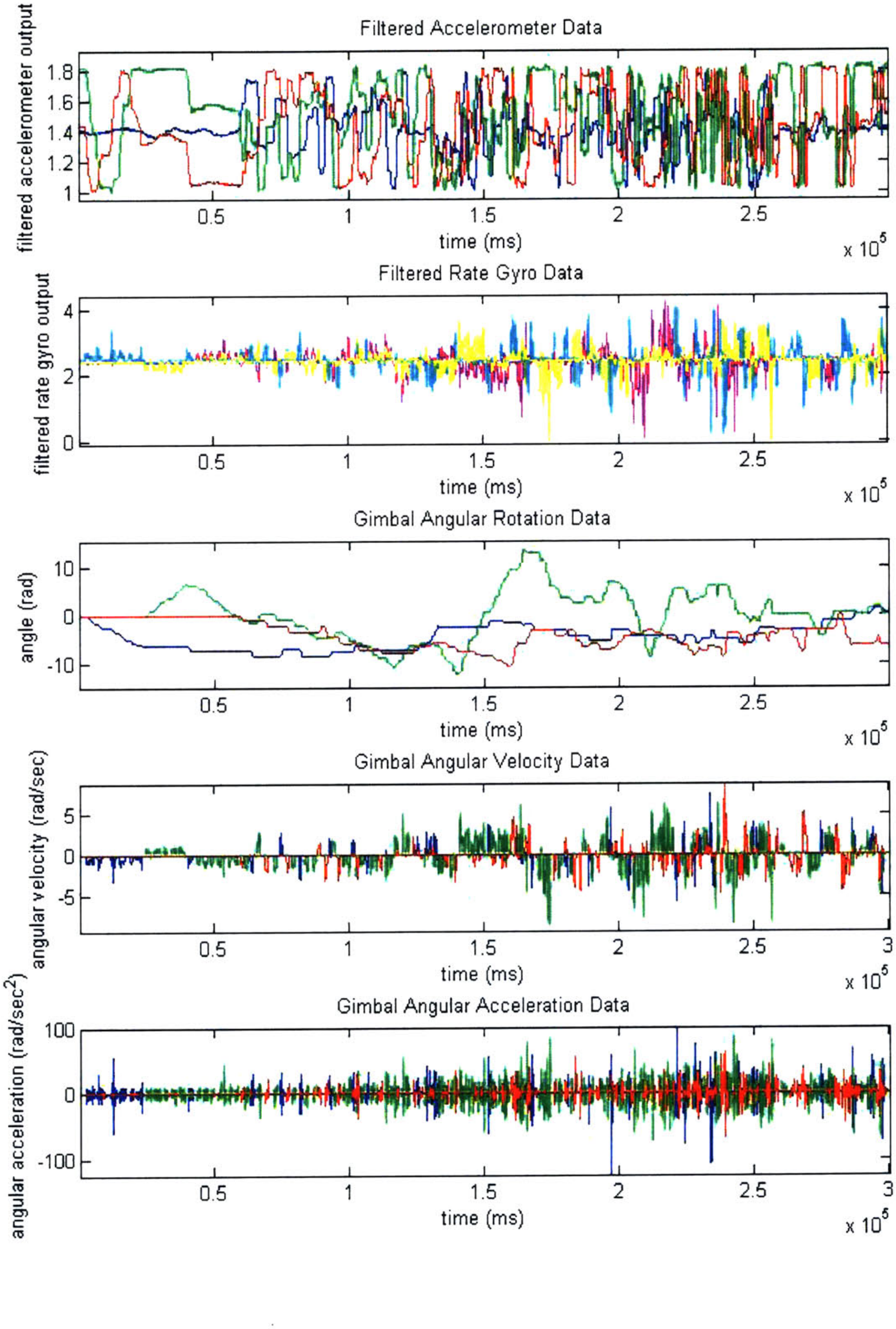

Figure 7-1: Accelerometer, Rate Gyro, Gimbal Angular Data (5 Minute Trial)

7.1 Accelerometer Parameters

Using the calibration algorithm described in Chapter 2, the accelerometer parameters are estimated, (Table 7.1).

Estimated Accelerometer Parameters From 180,000 Data Points

Sensor	Orientation $\widehat{\vec{S_{i\mathbf{s}}}}$			Scaling	Position $\widehat{\vec{D_{i\mathbf{s}}}}$			Distance	Offset
i	s_x	s_y	s_z	$\lvert\widehat{\vec{S_{i\mathbf{s}}}}\rvert$	d_x	d_y	d_z	$\lvert\widehat{\vec{D_{i\mathbf{s}}}}\rvert$	$\widehat{\mu_i}$
1	0.0008	-0.0396	-0.0001	0.0396	0.0050	-0.0201	-0.0004	0.0207	1.3973
2	0.0034	-0.0021	-0.0413	0.0415	0.0101	-0.0111	0.0010	0.0150	1.4148
3	0.0404	-0.0010	0.0007	0.0405	-0.0018	-0.0096	-0.0005	0.0097	1.4121

Table 7.1: Estimated Accelerometer Parameters ($\widehat{\vec{S_{i\mathbf{s}}}}$, $\lvert\widehat{\vec{S_{i\mathbf{s}}}}\rvert$, $\widehat{\vec{D_{i\mathbf{s}}}}$, $\lvert\widehat{\vec{D_{i\mathbf{s}}}}\rvert$, $\widehat{\mu_i}$)

The scaling for each of the accelerometers is around 0.04, which corresponds to 40mV per 1 m/sec^2, or 392mV per 1g, which is close to the 333mV per 1g listed in the STMicro specification sheet, [14] The offset voltage, which corresponds to 0 linear acceleration, is between 1.353V and 1.947V for all of the accelerometers in accordance with the specification sheet. $\lvert\vec{D_{i\mathbf{s}}}\rvert$, the distance from the center of each of the accelerometer sensor frames to the center of rotation, is roughly 2cm, 1.5cm, and 1cm. These distances appear reasonable with reference to the positioning of the 3-axis accelerometer chip on the IMU, the physical dimensions of the chip, and the positioning of the IMU inside the inner ring of the gimbal. Furthermore, the angular difference between each of the orientation axes of the accelerometers is close to 90 degrees, (Table 7.2), and within the 4 degree skew error referenced by the specification sheet.

Estimated Angle Between Accelerometers

$\angle(1,2)$	$\angle(1,3)$	$\angle(2,3)$
86.88°	87.35°	86.20°

Table 7.2: Estimated Angle Between Accelerometer Orientation Vectors, $\widehat{\vec{S_{i\mathbf{s}}}}$

7.1.1 Convergence of $\widehat{\vec{S}}_{i\mathrm{S}}$, $\widehat{\mu_i}$ and $\widehat{\vec{D}}_{i\mathrm{S}}$

Recall from Chapter 2 that the $(\widehat{\vec{S}}_{i\mathrm{S}}^{T}, \widehat{\mu_i})$ and $\widehat{\vec{D}}_{i\mathrm{S}}$ vectors are solved for iteratively, one after the other. After 3 iterations of the algorithm, the vector elements converge to within 1e-4. In particular, the position vectors, $\widehat{\vec{D}}_{i\mathrm{S}}$, converge to within 1/10 of a millimeter of their final values, (Table 7.3).

Convergence of $\widehat{\vec{S}}_{i\mathrm{S}}$, $\widehat{\mu_i}$ and $\widehat{\vec{D}}_{i\mathrm{S}}$

iteration		1	2	3
$\widehat{\vec{D}}_{1\mathrm{S}}$	x	0.0000	0.0050	0.0050
	y	0.0000	-0.0200	-0.0201
	z	0.0000	-0.0004	-0.0004
$\widehat{\vec{S}}_{1\mathrm{S}}$	x	0.0008	0.0050	0.0050
	y	-0.0396	-0.0200	-0.0201
	z	-0.0001	-0.0004	-0.0004
$\widehat{\mu_1}$		1.3972	1.3973	1.3973
$\widehat{\vec{D}}_{2\mathrm{S}}$	x	0.0000	0.0100	0.0101
	y	0.0000	-0.0110	-0.0111
	z	0.0000	0.0012	0.0010
$\widehat{\vec{S}}_{2\mathrm{S}}$	x	0.0034	0.0034	0.0034
	y	-0.0020	-0.0021	-0.0021
	z	-0.0414	-0.0413	-0.0413
$\widehat{\mu_2}$		1.4147	1.4148	1.4148
$\widehat{\vec{D}}_{3\mathrm{S}}$	x	0.0000	-0.0018	-0.0018
	y	0.0000	-0.0096	-0.0096
	z	0.0000	-0.0005	-0.0005
$\widehat{\vec{S}}_{3\mathrm{S}}$	x	0.0404	0.0404	0.0404
	y	-0.0010	-0.0010	-0.0010
	z	0.0007	0.0007	0.0007
$\widehat{\mu_3}$		1.4121	1.4121	1.4121

Table 7.3: Convergence of Accelerometer Offset, Orientation, Position Vectors

7.2 Rate Gyro Parameters

The scaling for each of the rate gyros is around 0.29, which corresponds to 290mV per 1 rad/sec, or 5.06mV per 1 deg/sec. The magnitudes of the estimated scalings

Estimated Rate Gyro Parameters From 180,000 Data Points

Rate Gyro	Orientation $\widehat{\vec{V_{i\mathrm{S}}}}$			Scaling	Offset
i	v_x	v_y	v_z	$\widehat{\lvert\vec{V_{i\mathrm{S}}}\rvert}$	$\widehat{\chi_i}$
1	-0.0239	-0.2891	-0.0004	0.2901	2.4550
2	-0.2920	0.0357	-0.0179	0.2947	2.4313
3	0.0164	0.0155	-0.2973	0.2982	2.4417

Table 7.4: Estimated Rate Gyro Parameters ($\widehat{\vec{V_{i\mathrm{S}}}}$, $\widehat{\lvert\vec{V_{i\mathrm{S}}}\rvert}$, $\widehat{\chi_i}$)

match with the typical rate gyro scaling of 5mV per 1 deg/sec as listed on the Analog Devices specification sheet [10]. The offset voltage, which corresponds to 0 angular velocity, falls within the typical range of 2.3V and 2.7V for each of the sensors. The angular difference between each of the orientation axes of the rate gyros is close to 90 degrees, (Table 7.5), and represents the skew angle between the sensors when mounted "orthogonally" by hand.

Estimated Angle Between Rate Gyros

$\angle(1,2)$	$\angle(1,3)$	$\angle(2,3)$
92.24°	93.15°	89.30°

Table 7.5: Estimated Angle Between Rate Gyro Orientation Vectors, $\widehat{\vec{V_{i\mathrm{S}}}}$

7.3 Accelerometer, Rate Gyro Output Predictions

The estimated accelerometer parameters and the angular acceleration and velocity measurements from the gimbal are used to estimate the accelerometer outputs.

$$\widehat{A_i[k]} = (\widehat{\vec{S}^T_{i\mathrm{S}}}, \widehat{\mu_i}) \cdot \begin{bmatrix} R_{W\to S}[k] \cdot (\vec{G} + f(\vec{\alpha_{\mathrm{W}}}[k], \vec{\omega_{\mathrm{W}}}[k]) \cdot R_{S\to W}[k] \cdot \widehat{\vec{D_{i\mathrm{S}}}}) \\ 1 \end{bmatrix} \tag{7.1}$$

At all levels of resolution, the sensor output estimates very closely track the filtered sensor output voltages, (Figure 7-2), both of which track the raw accelerometer signals. The mean squared error (MSE) of the accelerometers is 0.0107V, 0.0144V, and 0.0104V respectively, which correspond to average errors of roughly 27mg, 35mg, and 26mg respectively. These errors are greater than the 1.6mg predicted from the specification sheet for a sampling rate of 1000Hz. The errors qualitatively appear to increase during periods of IMU movement, which corroborates the Kalman filter's use of relying more on the rate gyros during periods of rotation.

The estimated rate gyro parameters and the angular velocity measurements from the gimbal are used to estimate the rate gyro outputs.

$$\widehat{Y_i[k]} = (\widehat{\vec{V}_{i\mathbf{S}}^T}, \widehat{\chi_i}) \cdot \begin{bmatrix} R_{W\to S}[k] \cdot \vec{\omega_{\mathbf{W}}}[k] \\ 1 \end{bmatrix} \tag{7.2}$$

Again, at all levels of resolution, the sensor output estimates very closely track the filtered sensor output voltages, both of which track the raw rate gyro signals, (Figure 7-3). The mean squared error (MSE) of the rate gyros is 0.0152V, 0.0110V, and 0.0131V respectively, which correspond to average errors of roughly 3.00 deg/sec, 2.14 deg/sec, and 2.52 deg/sec, respectively. These errors are within the 3.16 deg/sec predicted from the specification sheet for a sampling rate of 1000Hz.

7.4 Gravity, Angular Velocity Vector Predictions

The linear acceleration vector in the sensor reference frame, $\widehat{\vec{l_{\mathbf{S}}}}[k]$, differs for each of the accelerometers, because each sensor is positioned differently on the IMU, with a different $\vec{D_{i\mathbf{S}}}$. The projection of the gravity vector into the sensor reference frame, however, is the same for each of the sensors. Recall that the Kalman filter assumes the rotational and dynamic accelerations measured by the accelerometers are insignificant compared to the gravity vector. The estimated gravity vector in the sensor reference frame is approximated as the estimated linear acceleration in the sensor reference frame.

$$\widehat{\vec{g_{\mathbf{S}}}}[k] = \widehat{\vec{l_{\mathbf{S}}}}[k] \tag{7.3}$$

The estimated accelerometer parameters and the measured accelerometer output data are used to estimate the gravity vector in Equation (3.3). The estimated gravity

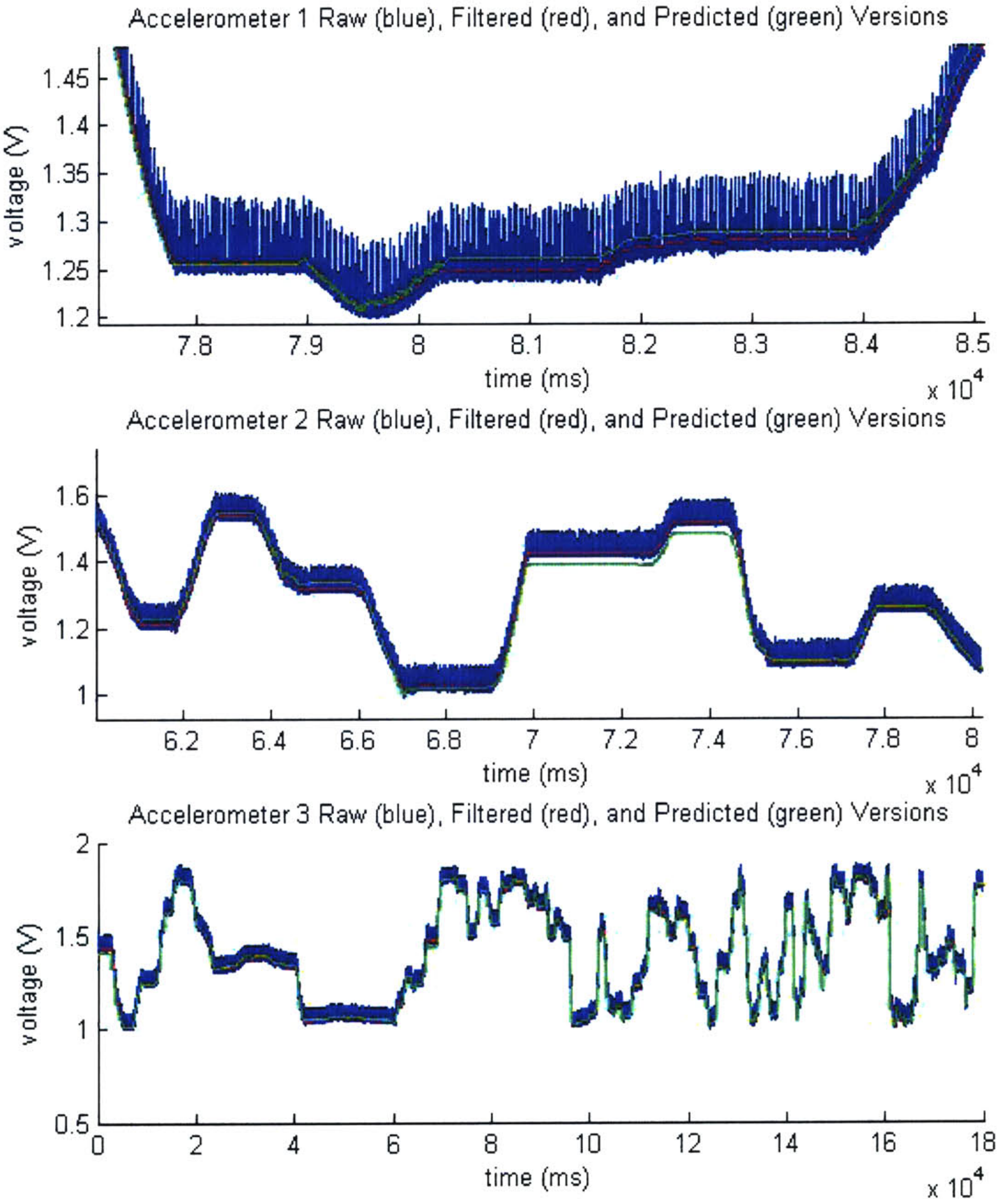

Figure 7-2: Raw(b), Filtered(r), Predicted(g) Accelerometer Values at Different Scales

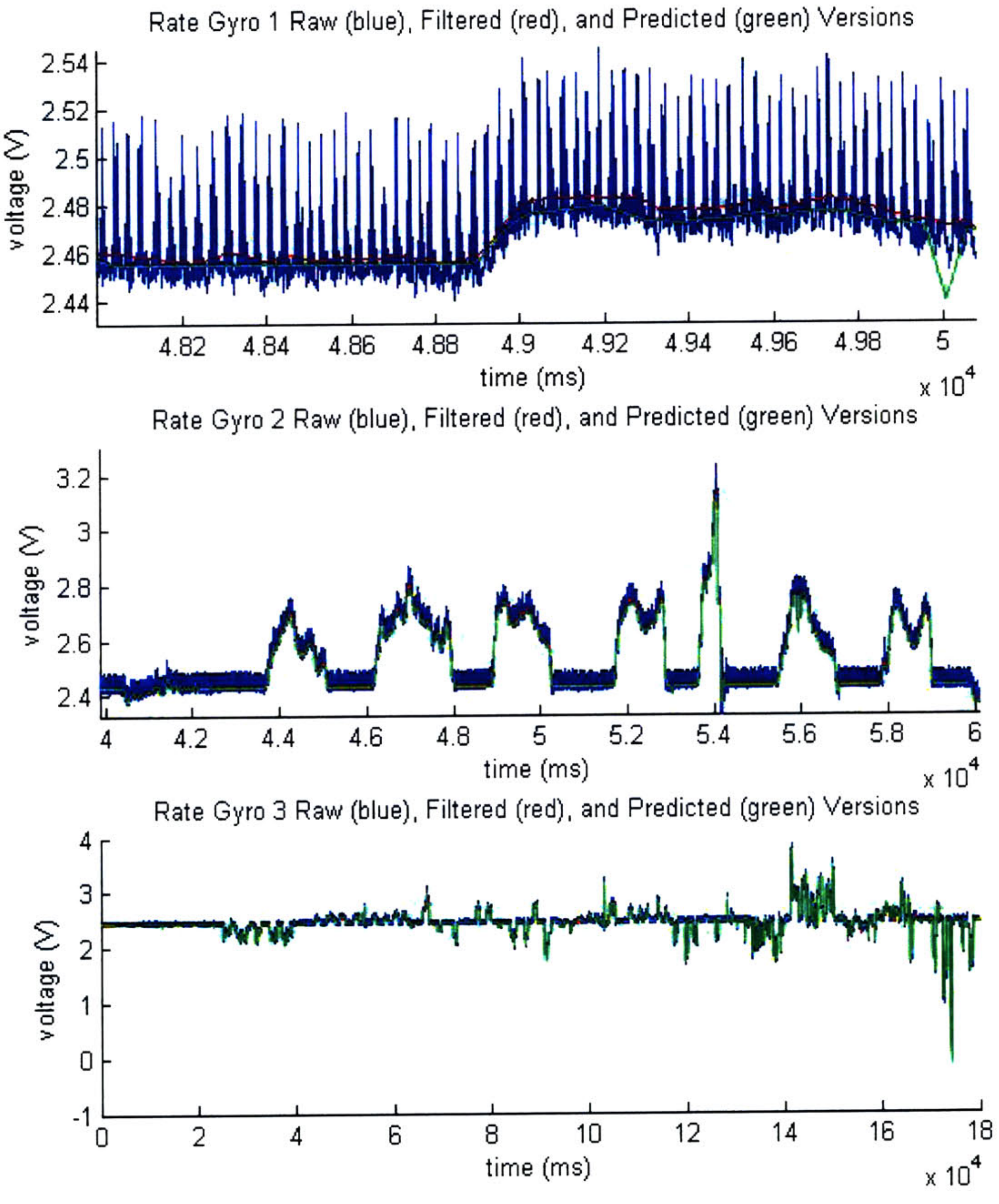

Figure 7-3: Raw(b), Filtered(r), Predicted(g) Rate Gyro Values at Different Scales

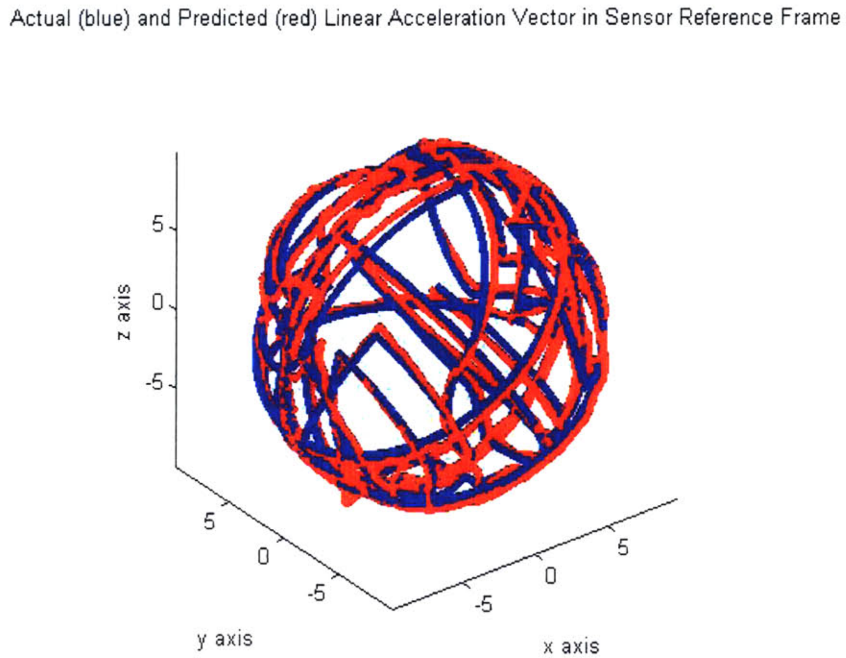

Figure 7-4: 3D Plot of Actual(b), Estimated(r) Gravity Vectors

vector qualitatively appears to track the actual gravity vector, (Figure 7-4). The MSE between the estimated and actual gravity vectors for the calibration data set is 0.4337 m/sec^2, which is roughly 4.5% of g. This large error in estimation is due to the assumption that gravity is the sole factor of acceleration, and is mitigated by the Hybrid Estimation in the Kalman filter.

The angular velocity vector in the sensor reference frame, w_S, is the same for each of the rate gyros, and is estimated from the estimated rate gyro parameters, and the measured rate gyro output data in Equation (3.5). The estimated angular velocity vector qualitatively appears to track the actual angular velocity vector, (Figure 7-5). The MSE between the estimated and actual angular velocity vectors for the calibration data set is 0.4337 rad/sec (24.85 deg/sec). This amount of error is large enough to accumulate resulting in a highly innaccurate orientation estimation, and requires the frequent periodic "reset" from the accelerometer information in the Kalman filter.

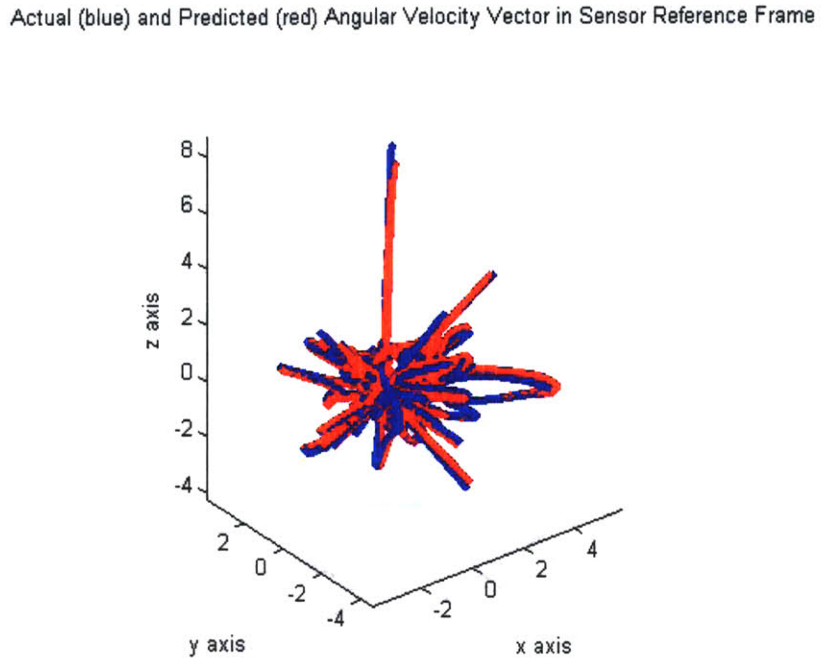

Figure 7-5: 3D Plot of Actual(b), Estimated(r) Angular Velocity Vectors

7.5 Kalman Estimate of YPR angles

The variance of the measurement noise, r, is calculated by dividing the average variance of the accelerometer noise, 3.3e-5V, by the average scaling factor of 0.0405 V per m/sec^2, resulting in r = 8.15e-4 m/sec^2. Similarly, the variance of the process noise, q, is calculated by dividing the average variance of the rate gyro noise, 1e-4V, by the average scaling factor of 0.294 V per rad/sec, resulting in q = 3.40e-4 rad/sec. The r and q values are used in the Kalman filter estimation of the YPR angles, (Figure 7-6). The estimated roll MSE is 1.7427 degrees and the variance is 2.4129 (standard deviation is 1.5534 degrees). The estimated pitch MSE is 3.1387 degrees and the variance is 15.7592 (standard deviation is 3.9698 degrees). The yaw estimate, which does not use the accelerometer information, and thus does not benefit from the Kalman filter, is so far from the actual angle that it often results in a non-sense, complex value. It should be noted that the roll angle measurement is larger than the pitch angle measurement because of the asymmetrical nature of the YPR representation.

In particular, the error used to estimate the pitch is carried over to the roll angle. Qualitatively, the majority of the errors occur during periods of little movement, which corresponds to not factoring in the accelerometer information enough against the rate gyro information.

7.6 Summary of Results

Overall, the calibration and YPR estimation algorithms are successful. The estimated offsets and scalings lie within the typical ranges documented in the specification sheets of the sensors. The position of the accelerometers on the IMU appear to match the placement of the accelerometer chip in the gimbal inner ring. The orientation of the sensors with respect to themselves is estimated as roughly orthogonal, as it should be, and the accelerometer skew is within the guidelines of the specification sheet. The estimated sensor output voltages track the actual sensor output voltages. The gravity vectors and angular acceleration vectors estimated from the IMU track the actual vectors derived from the gimbal. Using the calibrated parameters, and a raw dataset, the estimated YPR angles from the IMU sensor values tracks the angles (par yaw) derived from the gimbal rotation. The calibration technique works and can be refined to attain higher accuracy as noted in Chapter 8.

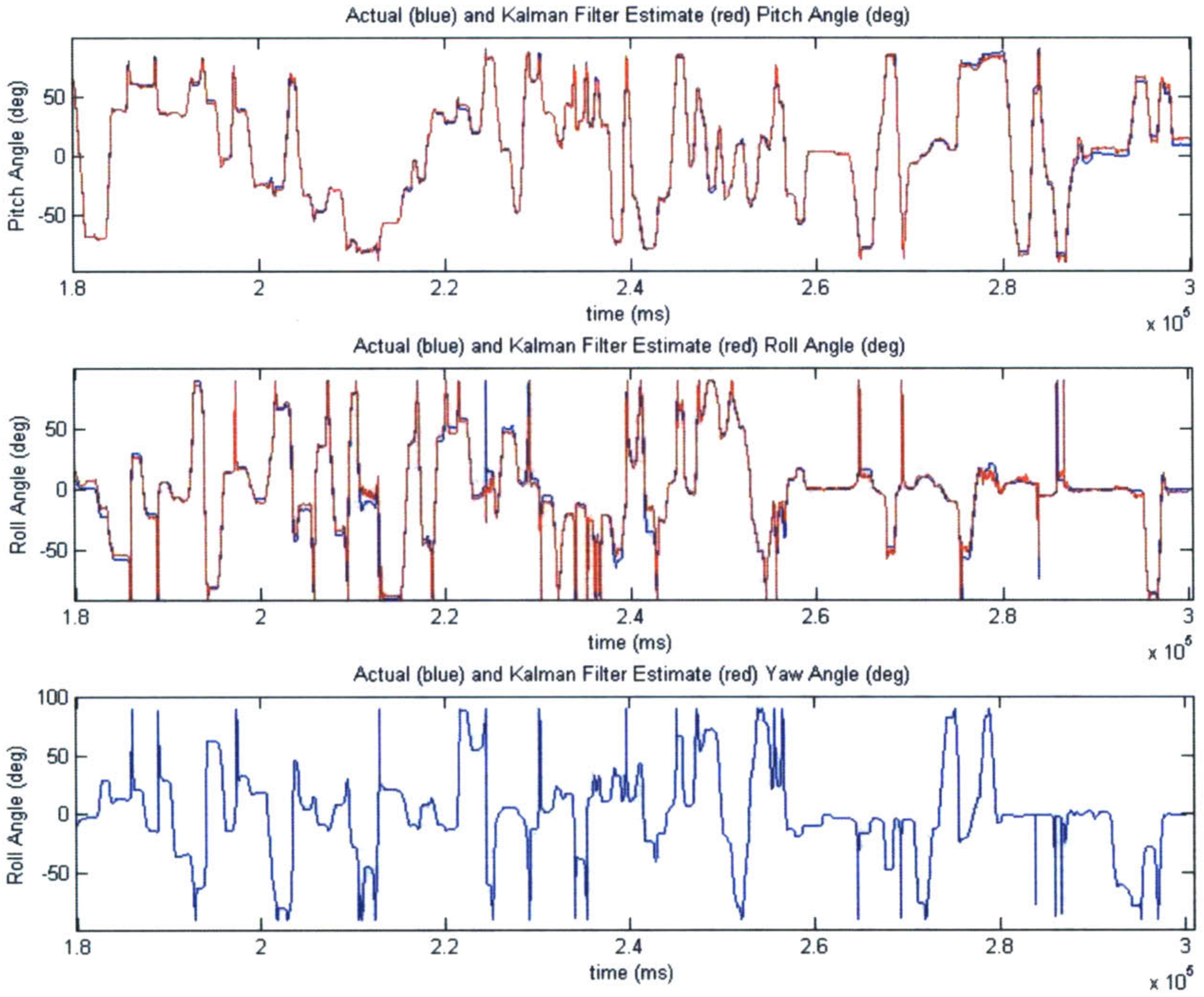

Figure 7-6: Kalman Estimation of Yaw, Pitch, Roll Angles

Chapter 8

Future Work

Many avenues can be taken to improve the method of calibrating an IMU using a gimbal. Some improvements involve a better user-interface, an improved calibration algorithm, and hardware upgrades/improvements. Eventually, the author hopes to apply the technology of calibrated IMUs with non-orthogonal components to the field of prosthetics and orthotics.

8.1 Calibration Improvements

The current state of the calibration algorithm is relatively rudimentary. The Matlab code is optimized to use as few *for* loops as possible, and to represent as much of the math in the form of direct matrix operations. The code could run faster if optimized and compiled in a different language, such as C.

8.1.1 Online Estimation

Currently, the batch estimator that is implemented for the calibration routine takes 2 minutes to estimate the IMU parameters given a 5 minute dataset. The estimator can be implemented efficiently in terms of a recursive estimator with a predictor-corrector structure as mentioned in Chapter 2. This will allow for continuous feedback to the user as to how the calibration parameters are converging over time.

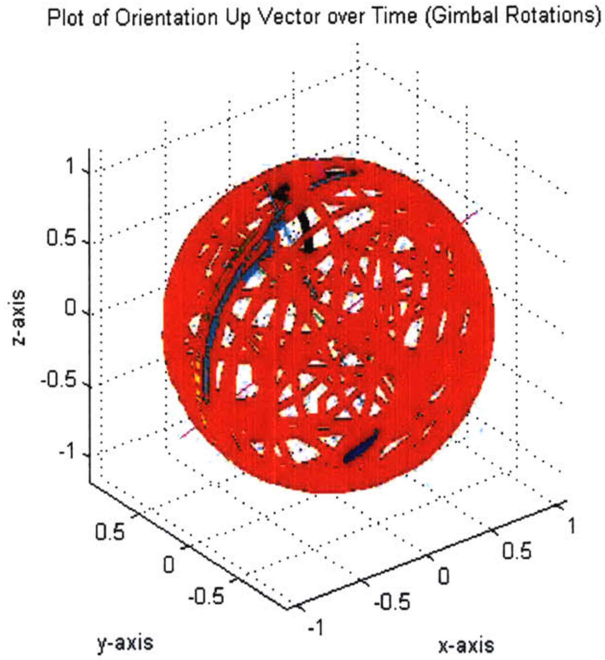

Figure 8-1: PCA of Orientation Up Vector Swept Through Time

8.1.2 Improved User Interface

A display of the orientations of the IMU swept through thus far would give feedback to prompt the user to sweep through unexplored orientations. The display would indicate which faces of the IMU have been swept through which orientations, (Figures 8-1, 8-2, 8-3). Furthermore, Principle Component Analysis (PCA) would identify which axis has been swept through most, and which axis needs to be swept through more. By selecting the eigenvectors of the covariance matrix of the orientation vectors, PCA can indicate how much variance exists along each principle component. This variance be used to scale the amount of information used from each swept region, so as not to overfit a particular region of data.

8.1.3 Goodness of Fit

Alternatively, instead of visualizing the orientation vectors sweeping through space, a metric should be devised with a score corresponding to the "goodness" of the

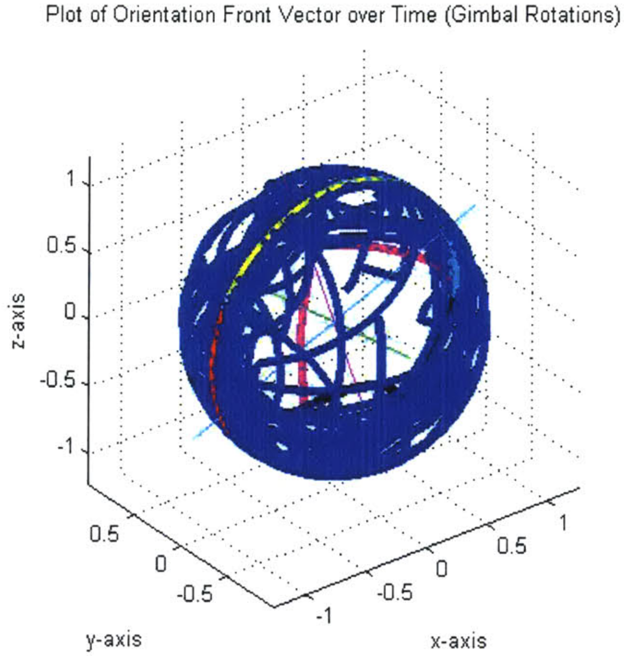

Figure 8-2: PCA of Orientation Front Vector Swept Through Time

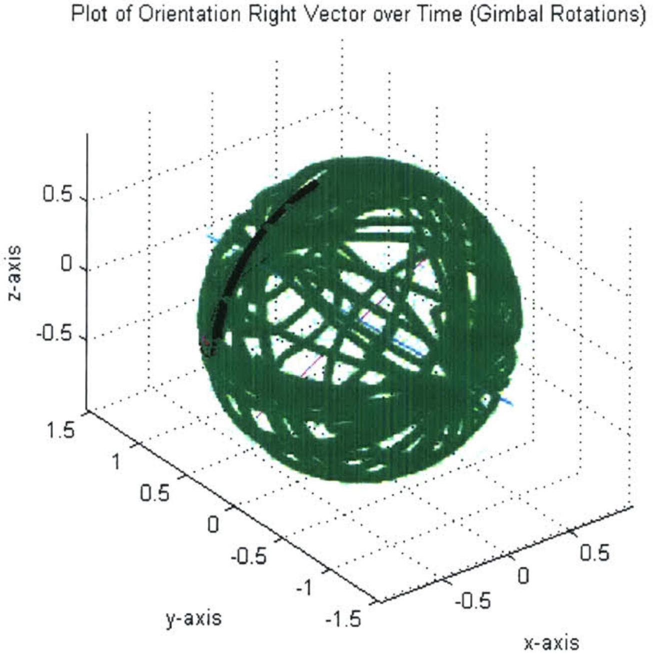

Figure 8-3: PCA of Orientation Right Vector Swept Through Time

input space. The covariance matrix of the elements of the rotation matrix, R can be compared to the profile of equally distributed rotation matrices which transform a vector equally about a sphere, to mathematically determine which Rs need to be focused on. An analog metric can be conceived by applying Radial Basis Functions to the surface of the sphere swept out by orientation vectors, [22].

More than checking to see that the IMU sweeps through all regions of space equally, the quality of good input data should be measured as how much of the space of angular velocity and linear acceleration the IMU sweeps through. The quality of the input data can be determined by categorizing the vectors of interest into spatially spherical histograms, with bins of equal steradian measure. Planar spherical histograms can be expanded to volumetric spherical histograms. One such instantiation would have k levels corresponding to radii and n, equally spaced points on each of the k spheres corresponding to bin centers of voxels in the spatial histogram. The n, equally spaced points can be iteratively determined, or an arbitrary spherically symmetric shape, such as a Buckyball, with n = 60, can be used.

8.2 Kalman Filter Improvement

The Kalman filter measurement and process noise gains, p and r, can be optimized, as well as the ϵ threshold for the hybrid estimation. In addition to tuning the gains, the implementation of the Kalman filter can be also be improved. As noted in Chapter 7, the majority of the errors appear to occur during periods of little to no movement. During these periods, the accelerometer information ought to be weighed more than it currently is. Instead of using a binary hybrid component, ρ, an analog ρ could act as a scaling factor that uses the accelerometer information inversely proportional to the period and magnitude of non-gravitational acceleration. Also, as noted in [23], the hybrid component ought to include a time-delay so as not to include accelerometer information during periods of acceleration fluctuation, in which the acceleration transitions through a state of having magnitudes equal to gravity. The time-delay will ensure that the accelerometer information is only used during periods of truly "low" acceleration.

A method of using the Kalman filter to estimate the yaw component (using only accelerometers and rate gyros) is considered. In particular, the estimated angular velocity vector, $\widehat{\vec{\omega}_{\mathbf{S}}}[k]$ is used to generate $S(\widehat{\vec{\omega}_{\mathbf{S}}}[k])$, which is used to generate the incremental rotation matrix, $R_{inc}[k]$, which is in turn used to estimate the gravity vector $\widehat{\vec{g}_{\mathbf{S}}}[k]$. Depending on the value of ϵ, the accelerometer information may affect the output of $\widehat{\vec{g}_{\mathbf{S}}}[k]$. It may be possible to re-estimate $\widehat{\vec{\omega}_{\mathbf{S}}}[k]$ such that a new $R_{inc}[k]$ is responsible for transforming $\widehat{\vec{g}_{\mathbf{S}}}[k]$ to $\widehat{\vec{g}_{\mathbf{S}}}[k+1]$. The new $\widehat{\vec{\omega}_{\mathbf{S}}}[k]$ could be used to increment the yaw angle.

8.3 IMU Improvements

Because the only sensors used in this implementation of an IMU are rate gyros and accelerometers, the yaw component has no static-field feedback. Magnetometers could be added to the IMU package to measure the magnetic field, which, while it changes based on latitude and is subject to EM interference from electronic equipment and rebar inside buildings, is roughly static. The magnetometer information can be tied into the Kalman filter equations to get a much better estimate of yaw angle, [18]. The magnetometers can be calibrated in exactly the same way as the accelerometers, hopefully with minimal EM interference from the gimbal and wiring.

A thermistor could be also be included in the IMU package to account for changes in the rate gyro's temperature-dependent offset voltage. The offset voltage could be implemented as a function of the temperature and solved for non-linearly (or estimated as a linear function), while still using the same equations laid out in Chapter 2.

8.4 Gimbal Improvements

The current version of the gimbal uses audio stereo connectors as axes of rotation. The stereo connectors should be replaced with sliprings which are meant to take the load of the gimbal rings and continuous rotation. The axes are pressfit and glued in place which may account for errors due to non-colinear axes of rotation. A precision machined endstop would allow for the correct depth setting on the dowel pins, and would lead to a more precise alignment of the rotation axes.

8.4.1 AutoCalibration

Eventually, the human can be eliminated from the calibration loop. Given appropriate feedback about which orientations and which angular velocities and accelerations need to be swept, stepper motors can actuate each of the gimbal axes. In this manner, higher angular velocities and accelerations may be used to calibrate the sensors. Also, the fully automated calibration routine would calibrate IMU sensors in an optimal amount of time.

8.4.2 Translation

Currently, the calibration procedure only accounts for orientation in space. It may be possible to induce linear accelerations by mounting the gimbal in an assembly that can translate in space. One manifestation of this assembly would include 3 linear actuators attached to platforms, nested one inside the other. The gimbal would sit inside the inner platform. In this manner, 3-axes of translation and 3-axes of rotation can be applied simultaneously.

8.5 Data Processing Improvements

Currently, the calibration process occurs offboard of the PC104 and on a host computer. The first step toward a standalone calibration setup would be to implement the real-time calibration algorithm on the PC104 itself in a stand-alone mode. The end goal would be to implement the calibration algorithm as well as filtering and Kalman estimation on an FPGA or microcontroller attached to the IMU, so each IMU can store its own calibration information, and directly output YPR angles.

Chapter 9

Conclusion

In this thesis, a novel method for calibrating IMUs using a gimbal is developed. The IMU is comprised of 3-axes of accelerometers and 3-axes of rate gyros. A calibration algorithm is developed for estimating the sensor parameters from known world quantities taken from the gimbal. A Kalman filter is implemented to estimate the roll and pitch of the IMU. The physical hardware for designing and assembling an IMU and the gimbal is performed with incremental revisions. The electrical noise profile of the IMU sensors is modeled as AWGN noise, with the variance of the accelerometer noise estimated at 3.3e-5V and the the variance of the rate gyro noise estimated at 1e-4V. An appropriate zero-phase-lag MAW filter with window length of 50, is chosen to compensate for IMU noise and smooth gimbal discretization. The sensor parameters are calibrated, and then used to estimate YPR angles using the Kalman filter. The estimated pitch MSE is 1.7427 degrees. The estimated roll MSE is 3.1387 degrees. The estimated yaw MSE, which does not benefit from the accelerometer data, nor the Kalman filter, is too large to measure. A series of improvements in the hardware design, the calibration algorithm, and the user interface are sought in the future. These improvements will lead to more accurate estimations of YPR angles, as well as a faster calibration process. The appendices contain schematics and printed circuit board (PCB) layouts of the IMU, engineering drawings of the gimbal, and Matlab code used to calibrate the sensors and estimate RPY from the IMU.

Bibliography

[1] Ari Benbaset. An inertial measurement unit for user interfaces. Master's thesis, Massachusetts Institute of Technology, 2000.

[2] Paul Bourke. http://astronomy.swin.edu.au/ pbourke/geometry/rotate/ rotate a point about an arbitrary axis (3 dimensions). accessed: 8/31/05.

[3] Robert Brown and Patrick Hwang. *Introduction to Random Signals and Applied Kalman Filtering*, chapter 1.2,3,10. John Wiley & Sons, 1985.

[4] Olivier Cadet. Introduction to kalman filtering and its use in dynamic positioning systems. *Dynamic Positioning Conference*, 2003.

[5] Imre Cikajlo, Zlatko Matjacic, Tadej Bajd, and Nozomu Hoshimiya. The use of kalman filtering in assistive device for data assessment and control in gait re-education. *International Converence on Intelligent Robots and Systems*, 2004.

[6] Rongching Dai, Richard Stein, Brian Andrews, Kelvin James, and Marguerite Wieler. Application of tilt sensors in functional electrical stimulation. *IEEE Transactions on Rehabilitation Engineering*, 4(2):63–72, 1996.

[7] John Hall, Robert Williams II, and Frank Grass. Cartesian control for the inertial measurement unit calibration platform. Technical report, Ohio University, 2000.

[8] http://en.wikipedia.org/wiki/Inertial_guidance_system. accessed: 8/31/05.

[9] http://ocw.mit.edu/NR/rdonlyres/Aeronauticsand Astronautics/16333Fall-2004/lecture_15.pdf. Inertial sensors, complementary filtering, simple kalman filtering. accessed: 8/31/05.

[10] ADXRS30000.html http://www.analog.com/en/prod/0 2877˙ Analog devices 3-axis rate gyro specification sheet. accessed: 8/31/05.

[11] http://www.fisica.uniud.it/ ercolessi/md/md/node22.html. Predictor-corrector structure. accessed: 8/31/05.

[12] http://www.hq.nasa.gov/office/pao/History/alsj/4gimb.jpg. Apollo 4-axis gimbal. accessed: 8/31/05.

[13] http://www.hq.nasa.gov/office/pao/History/alsj/e 1344.htm. Apollo lunar surface journal. accessed: 8/31/05.

[14] http://www.st.com/stonline/products/literature/ds/10221/lis3l02as4.pdf. Stmicro 3-axis accelerometer specification sheet. accessed: 8/31/05.

[15] Albert Krohn, Michael Beigl, Christian Decker, Uwe Kochendorfer, Philip Robinson, and Tobias Zimmer. Inexpensive and automatic calibration for acceleration sensors. Technical report, Universitat Karlsruhe, 2000.

[16] Henk Luinge and Peter Veltink. Inclination measurement of human movement using a 3-d accelerometer with autocalibration. *IEEE Transactions on Neural Systems and Rehabilition Engineering*, 12(1):112–121, 2004.

[17] Paul Lukowicz, Holger Junker, and Gerhard Troster. Automatic calibration of body worn acceleration sensors. Technical report, Institute for Computer Systems and Networks, UMIT Innsbruck, Austria.

[18] Joao Marins, Xiaoping Yun, Eric Bachmann, Robert McGhee, and Michael Zyada. An extended kalman filter for quaternion-based orientation estimationg using marg sensors. *Proceedings of the International Conference on Intelligent Robots and Systems*, 2001.

[19] R. McGhee, E. Bachmann, X. Yun, and M. Zyda. Rigid body dynamics, inertial reference frames, and graphics coordinate systems: A resolution of conflicting conventions and terminology. Technical report, Naval Postgraduate School.

[20] R. McGhee, E. Bachmann, X. Yun, and M. Zyda. Real-time tracking and display of human limb segment motions using sourceless sensors and a quaternion-based filtering algorithm- part 1: Theory. *Presence*, 2001.

[21] Aboelmagd Noureldin, Ahmed Osman, and Naser El-Sheimy. A neuro-wavelet method for multi-sensor system integration for vehicular navigation. *Measurement Science and Computing*, 15:404 412, 2004.

[22] Mark Orr. Introduction to radial basis function networks. Technical report, Centre for Cognitive Sciences, University of Edinburgh, Scotland.

[23] Henrik Rehbinder and Xiaoming Hu. Drift-free attitude estimation for accelerated rigid bodies. *Journal of Automatica*, 2003.

[24] P. Veltink, P. Slycke, J. Hemssems, R. Buschman, G. Bultstra, and H Hermens. Three dimensional inertial sensing of foot movements for automatic tuning of a two-channel implantable drop-foot stimulator. *Medical Engineering & Physics*, 2003.

[25] Peter Veltink, Henk Luinge, Bart Kooi, Chris Baten, and Per Slycke. The artificial vestibular system- design of a tri-axial inertial sensor system and its applications in the study of human environment. Technical report, Institute for Biomedical Technology (BMTI) and MESA Institute, University of Twente, Netherlands.

[26] Antoon Willemsen, Carlo Frigo, and Herman Boom. Drift-free attitude estimation for accelerated rigid bodies. *IEEE Transactions on Biomedical Engineering*, 38(12):1186–1197, 1991.

[27] www.animazoo.com. accessed: 8/31/05.

[28] www.gyration.com. accessed: 8/31/05.

[29] www.intersense.com/products/prec/ic3/wirelessic3.htm. accessed: 8/31/05.

[30] www.memsense.com. accessed: 8/31/05.

[31] www.microstrain.com. accessed: 8/31/05.

[32] www.silicondesigns.com. accessed: 8/31/05.

[33] www.xbow.com. accessed: 8/31/05.

[34] www.xsens.com. accessed: 8/31/05.

[35] Rong Zhu and Zhaoyinbg Zhou. A real-time articulated human motion tracking using tri-axis inertial/magnetic sensors package. *IEEE Transactions on Neural Systems and Rehabilition Engineering*, 12(2):295–302, 2004.