GYROLESS NANOSATELLITE ATTITUDE DETERMINATION USING AN ARRAY OF SPATIALLY DISTRIBUTED ACCELEROMETERS

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Aerospace Engineering

by

Kory J. Haydon

June 2023
COMMITTEE MEMBERSHIP

TITLE: Gyroless Nanosatellite Attitude Determination Using an Array of Spatially Distributed Accelerometers

AUTHOR: Kory J. Haydon

DATE SUBMITTED: June 2023

COMMITTEE CHAIR: Eric Mehiel, Ph.D.
Professor of Aerospace Engineering at Cal Poly, SLO

COMMITTEE MEMBER: Leonardo Torres, Ph.D.
Professor of Electronic Engineering at UFMG

COMMITTEE MEMBER: Kira Abercromby, Ph.D.
Professor of Aerospace Engineering at Cal Poly, SLO

COMMITTEE MEMBER: John Bellardo, Ph.D.
Professor of Computer Science at Cal Poly, SLO
ABSTRACT

Gyroless Nanosatellite Attitude Determination Using an Array of Spatially Distributed Accelerometers

Kory J. Haydon

The low size and budget of typical nanosatellite missions limit the available sensors for attitude estimation. Relatively high noise MEMS gyroscopes often must be employed when accurate knowledge of the spacecraft’s angular velocity is necessary for attitude determination and control. This thesis derived and tested in simulation the “Virtual Gyroscope” algorithm, which replaced a standard gyroscope with an array of spatially distributed accelerometers for a 1U CubeSat mission. A MEMS accelerometer model was developed and validated using Root Allan Variance, and the Virtual Gyroscope was tested both in the open loop configuration and as a replacement for a gyroscope in a Multiplicative Extended Kalman Filter. It was found that the quality of the Virtual Gyroscope’s rate measurement improved with a larger and higher quality array, but the error in the estimate was very large. The low signal-to-noise ratio and the unknown bias in the accelerometers caused the angular velocity estimate from the accelerometer array to be too poor for use in the propagation step of the Kalman filter. The Kalman filter performed better with attitude measurements alone than with the Virtual Gyroscope, even when the attitude were delivered at a low rate with added noise. Overall, the current Virtual Gyroscope algorithm that is presented in this thesis is not suitable to replace a MEMS gyroscope in a nanosatellite mission, although there is room for future improvements using bias prediction for the individual accelerometers in the array.
ACKNOWLEDGMENTS

Thanks to:

- My wonderful parents who taught me to love learning and chase my dreams. Also thank you for proofreading my writing, I’m sorry you had to suffer so.

- Dr. Leo Torres for his indispensable tutelage and advice in the research and writing of this thesis.

- Andrew Guenther for uploading this LaTeX template.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
<th>ix</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>x</td>
</tr>
<tr>
<td>NOMENCLATURE</td>
<td>xii</td>
</tr>
</tbody>
</table>

## CHAPTER

1. Introduction

1.1 Research Questions

1.2 Background and Motivation

1.2.1 Key Literature

1.3 Methodology

2. Deriving the Virtual Gyroscope

2.1 Accelerometer and Virtual Gyroscope Measurement Models

2.2 Solution without Noise or Bias

2.3 Solution with White Noise

2.4 Solution with White Noise and Bias

2.5 Chapter 2 Conclusion

3. Simulating the Virtual Gyroscope

3.1 Simplified Satellite Dynamics Model

3.2 Accelerometer Model

3.2.1 Accelerometer Error Sources

3.2.1.1 Stochastic Error Sources

3.2.1.2 Deterministic Error Sources

3.2.1.3 Final Mathematical Model
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.2</td>
<td>Stochastic Noise Model Validation: ADXL 357</td>
<td>33</td>
</tr>
<tr>
<td>3.3</td>
<td>Array Configurations</td>
<td>36</td>
</tr>
<tr>
<td>3.4</td>
<td>Testing the Virtual Gyroscope</td>
<td>39</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Performance Metrics</td>
<td>41</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Test Case A: Zero Input Without Initial Bias</td>
<td>43</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Test Case B: Constant Nonzero Input Without Initial Bias</td>
<td>50</td>
</tr>
<tr>
<td>3.4.4</td>
<td>Test Case C: Variable Angular Velocity Without Initial Bias</td>
<td>52</td>
</tr>
<tr>
<td>3.4.5</td>
<td>Test Case A, B, and C Conclusions</td>
<td>53</td>
</tr>
<tr>
<td>3.4.6</td>
<td>Test Case D: Variable Temperature</td>
<td>54</td>
</tr>
<tr>
<td>3.4.7</td>
<td>Test Case E: Initial Bias</td>
<td>55</td>
</tr>
<tr>
<td>3.5</td>
<td>Chapter 3 Conclusions</td>
<td>57</td>
</tr>
<tr>
<td>4</td>
<td>Using the Virtual Gyroscope for Attitude Estimation</td>
<td>58</td>
</tr>
<tr>
<td>4.1</td>
<td>Attitude Sensor Measurement Model</td>
<td>59</td>
</tr>
<tr>
<td>4.2</td>
<td>Extended Kalman Filter Formulation</td>
<td>61</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Multiplicative Quaternion Error</td>
<td>62</td>
</tr>
<tr>
<td>4.2.1.1</td>
<td>Step 1: Initialize</td>
<td>63</td>
</tr>
<tr>
<td>4.2.1.2</td>
<td>Step 2: Compute Kalman Gain</td>
<td>64</td>
</tr>
<tr>
<td>4.2.1.3</td>
<td>Step 3: Update Using the New Measurements</td>
<td>64</td>
</tr>
<tr>
<td>4.2.1.4</td>
<td>Step 4: Propagate in Discrete Time</td>
<td>65</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Resolving Sign Ambiguity</td>
<td>67</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Measurement and Process Noise Covariance</td>
<td>68</td>
</tr>
<tr>
<td>4.2.4</td>
<td>Performance Metrics</td>
<td>69</td>
</tr>
<tr>
<td>4.3</td>
<td>Testing the Attitude Filter with Rapidly Available Attitude Measurements</td>
<td>74</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Accelerometer Quantity</td>
<td>74</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Default Parameters</td>
<td>40</td>
</tr>
<tr>
<td>3.2</td>
<td>Test Case A: Initial Conditions</td>
<td>44</td>
</tr>
<tr>
<td>3.3</td>
<td>Test Case A: Results</td>
<td>48</td>
</tr>
<tr>
<td>3.4</td>
<td>Test Case B: Initial Conditions</td>
<td>50</td>
</tr>
<tr>
<td>4.1</td>
<td>Nominal Array Configuration</td>
<td>69</td>
</tr>
<tr>
<td>4.2</td>
<td>$3\sigma$ Error for Various Array Sizes</td>
<td>74</td>
</tr>
<tr>
<td>4.3</td>
<td>$3\sigma$ Error for Various Noise Coefficients</td>
<td>75</td>
</tr>
<tr>
<td>4.4</td>
<td>$3\sigma$ Angle Error for VG and MEMS</td>
<td>84</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>8</td>
</tr>
<tr>
<td>2.2</td>
<td>9</td>
</tr>
<tr>
<td>3.1</td>
<td>26</td>
</tr>
<tr>
<td>3.2</td>
<td>34</td>
</tr>
<tr>
<td>3.3</td>
<td>35</td>
</tr>
<tr>
<td>3.4</td>
<td>36</td>
</tr>
<tr>
<td>3.5</td>
<td>37</td>
</tr>
<tr>
<td>3.6</td>
<td>38</td>
</tr>
<tr>
<td>3.7</td>
<td>45</td>
</tr>
<tr>
<td>3.8</td>
<td>46</td>
</tr>
<tr>
<td>3.9</td>
<td>47</td>
</tr>
<tr>
<td>3.10</td>
<td>49</td>
</tr>
<tr>
<td>3.11</td>
<td>51</td>
</tr>
<tr>
<td>3.12</td>
<td>53</td>
</tr>
<tr>
<td>3.13</td>
<td>54</td>
</tr>
<tr>
<td>3.14</td>
<td>56</td>
</tr>
<tr>
<td>4.1</td>
<td>70</td>
</tr>
<tr>
<td>4.2</td>
<td>71</td>
</tr>
<tr>
<td>4.3</td>
<td>71</td>
</tr>
</tbody>
</table>
4.4 Filtered Angle Error .............................................. 72
4.5 Filtered Angular Velocity ......................................... 73
4.6 Filter Performance with Variable Temperature ................. 76
4.7 Effect of Virtual Gyroscope on Attitude Filter Performance .... 78
4.8 Estimating Angular Velocity with the Bias Update .............. 79
4.9 Effect of Increased $\sigma_R$ ......................................... 80
4.10 Less Frequent Attitude Measurements with More Trust in VG . . 82
4.11 Less Frequent Attitude Measurements with Less Trust in VG . . 83
NOMENCLATURE

A non-exhaustive list of the most significant symbols and notations.

ω - Angular velocity vector true, measured, and estimated

α - Angular acceleration

η - Additive noise to a measurement

β - Additive bias to a measurement

a_i - Acceleration experienced by accelerometer i true and measured

σ and σ^2 - Standard deviation and variance

r_i - The position of an accelerometer in the body frame

x - The state vector of a system

u - The unknown vector for least squares z = Au

A - Block matrix of sensor positions for least squares z = Au

z - Known vector of acceleration differences for least squares z = Au

L - The fourth differencing matrix for accelerometer pairing

e_i - The residual corresponding to sensor i

l - The side length of a cube

F_c - Control force
$T_c$ - Control torque

$m$ - Mass

$I$ - The inertia matrix

$q$ - The attitude quaternion

$\epsilon$ - The vector component of the quaternion

$q_4$ - The scalar component of the quaternion

$\mathbb{I}_n$ - The $n \times n$ identity matrix

$Q$ - The white noise coefficient

$K$ - The random walk coefficient

$B$ - The bias instability coefficient

$S$ - Power spectral density

$f$ - Frequency

$T$ - Period or temperature by context

$k$ - A discrete time step

$t$ - Time

$y$ - MEKF measurement vector

$A(q)$ - The attitude matrix for quaternion $q$

$p$ - A physical vector with known components in inertial coordinates
$b$ - The vector $p$ expressed in body coordinates

$0_{n \times n}$ - An $n \times n$ matrix of zeros

$P_k$ - Error Covariance matrix

$R$ - Measurement noise covariance matrix

$Q_k$ - Process noise covariance matrix

$H_k$ - Sensitivity matrix

$K_k$ - Kalman Gain

Common Modifiers

$\tilde{v}$ - The measurement of vector $v$

$\dot{v}$ - The estimate of vector $v$

$\dot{\dot{v}}$ - The time derivative of vector $v$

$v^\times$ - The cross product matrix for vector $v$

$\Delta v_{i,j}$ - The difference between vectors $v_i$ and $v_j$

$v_m$ - The $m$th component of vector $v$

$v^T$ - The transpose of vector or matrix $v$

$v^{-1}$ - The inverse of matrix $v$

$v_0$ - The initial value of $v$

$v^-$ - A priori value of $v$
$v^+$ - A posteriori value of $v$

$|v|$ - Component-wise absolute value of $v$

$||v||$ - Magnitude of $v$
Chapter 1

INTRODUCTION

Chapter 1 discusses the topic and motivation for this thesis. In section 1.1, the main research question is introduced, as well as several secondary research questions that emerged in the course of study. These questions are then motivated in section 1.2, which establishes context and presents the key publication that this thesis seeks to build upon. Section 1.3 introduces the methodology chosen to attempt to answer the primary and secondary research questions and describes the structure of this thesis.

1.1 Research Questions

The primary research question for this thesis asks, can an array of spatially distributed accelerometers replace a gyroscope for angular velocity measurement in nanosatellite attitude determination? More specifically, what accelerometer array specifications and supporting filters are required for an array of accelerometers to meet or exceed the performance of a gyroscope in the context of nanosatellite operations?

The process of answering this question raised several additional queries:

- What methods are typically used to characterize the performance of attitude determination sensors and algorithms and how can they be applied to the accelerometer array?

- How can an accelerometer and a gyroscope be accurately simulated based on the performance characteristics provided by the sensor’s manufacturer?
Can the estimate of angular velocity from an accelerometer array directly replace the estimate from a gyroscope in well-established attitude estimation algorithms or are additional modifications to the filter warranted?

1.2 Background and Motivation

Many spacecraft applications require knowledge of the spacecraft’s attitude with respect to some reference frame for mission operations. Spacecrafts with large budgets and greater physical size can use expensive sensors such as star trackers to determine their attitude with great accuracy. The estimate of the spacecraft’s attitude obtained from a star tracker can then be optimally combined with an estimate of attitude using the spacecraft’s integrated angular velocity in a recursive filter such as a Multiplicative Extended Kalman Filter (MEKF) to obtain an even more accurate attitude estimate [7]. For larger missions, high accuracy (and high cost) gyroscopes such as fiber optic gyroscopes (FOGs) provide a low noise and low bias drift estimate of angular velocity to be used in the recursive filter.

Nanosatellites are defined as spacecrafts with mass between 1 and 10 kg [13]. These small satellites often adhere to the CubeSat form factor, which discretizes standard volume constraints for easier compatibility with launch providers. A 1U CubeSat, for example, has a volume constraint of a 10 cm cube and typically weighs less than 3 kg [2]. On this scale, satellite platforms become much more accessible for low budget research or commercial applications. Nanosatellites tend to have much lower development time and less rigorous launch qualification requirements, especially if they adhere to standard bus configurations that launch providers are familiar with. Because of these factors, more and more universities, research groups, and companies are building and launching nanosatellites for a wide range of applications. As of late
2022, 1897 CubeSats were launched, and even more nanosatellites were launched that did not follow the CubeSat standard [13].

![Figure 1: Annual CubeSat Launches](image)

**Figure 1.1: Annual CubeSat Launches [13].**

This number is only expected to increase.

With CubeSat’s accessibility comes some drawbacks. CubeSat size and budget constrains the available sensors. For attitude sensors, CubeSats often must rely on magnetometers, horizon sensors, and sun sensors which exhibit much lower performance than state-of-the-art star trackers [19]. Furthermore, angular rates are provided by relatively inexpensive microelectromechanical system (MEMS) gyroscopes which have much more noise and bias drift than FOGs [15].

With these limitations, the way is open for new attitude determination solutions. MEMS accelerometers are appealing sensors to use because they are very small, inexpensive, and exhibit much less bias drift than MEMS gyroscopes [17]. The idea
of estimating angular velocity with many accelerometers is not new to the aerospace community, but it has yet to be applied for CubeSat applications.

### 1.2.1 Key Literature

The problem of estimating angular velocity using an array of accelerometers was considered in a 2022 paper by Chris Hamley and John Crassidis titled "Angular Velocity Estimation and Error-Covariance Analysis from Accelerometers Using Least Squares" [11]. In this publication, Hamley and Crassidis consider the problem of estimating angular velocity from an array of accelerometers with Gaussian white noise but no measurement bias. It was found that the centripetal acceleration measured by the accelerometers due to the rotation of a vehicle can be used to estimate the size of the angular velocity components, but there will be inherent ambiguity in the sign of the angular velocity. Multiple least squares style solutions were presented and tested, but the simplification of biasless sensors, constant temperature, and a much larger vehicle than a CubeSat left plenty of room for further development. The paper also suggested a method for resolving the sign ambiguity of the measurement using an estimate of the angular acceleration of the vehicle, although this would be less effective for biased accelerometers.

This research also required a method to assess the performance of attitude estimation algorithms and gyroscopes. For the former, an effective strategy is to observe the residual between the estimate of the attitude and the true attitude, which is readily available if the algorithm is tested in a simulation. The algorithm is deemed effective if the residual converges within some desired bound around the true value. It is less instructive to examine the bounds of the residual of a gyroscope because bias drift will cause the estimate to diverge from the true value, especially over long time spans. This is why many attitude estimation algorithms that use gyroscopes must predict the
sensor’s bias and subtract it from the measurement at every filter time step. Because the noise and bias drift are stochastic in nature, the Root Allan Variance curve can be used to characterize noise for accelerometers and gyroscopes. The research paper Analysis and Modeling of Inertial Sensors Using Allan Variance by El-Sheimy et al. [10] provides an overview of the mathematics behind the Root Allan Variance curve and how it can be used to assess sensor performance. A more detailed treatment of the mathematics behind Root Allan Variance curve as discussed in [10] can be found in Appendix A.

1.3 Methodology

Building off the work from Hamley and Crassidis in [11], the main objective of this thesis is to build a “Virtual Gyroscope” algorithm that can replace a conventional gyroscope in attitude estimation algorithms. To limit this research to a reasonable scope, all experiments were conducted using MATLAB and Simulink simulations; no physical components were used. Additionally, analysis was constrained to analyzing the Virtual Gyroscope for use on a 1U CubeSat. The structure of the remainder of this thesis is as follows.

Chapter 2 derives the “Virtual Gyroscope” algorithm to estimate unsigned angular velocity of a CubeSat from an array of three-axis accelerometers. With help from [11], the algorithm was designed to input an arbitrary number of accelerometers at arbitrary locations in the spacecraft body. Based on common measurement models for an accelerometer, a measurement error model of the Virtual Gyroscope (VG) was then determined as a function of the measurement error for the individual accelerometer measurements. This measurement model was then compared to the measurement model for a MEMS gyroscope that is typically used in attitude estimation algorithms.
In Chapter 3, a MATLAB and Simulink model was created to simulate the 1U Cube-Sat’s rigid body dynamics, the array of accelerometers, and the VG algorithm. The VG performance was tested in the same way conventional gyroscopes are characterized by studying the angular velocity residual and computing a Root Allan Variance curve. This chapter also describes the mathematical model and procedure used to simulate a MEMS accelerometer based on a manufacturer’s datasheet for the sensor.

In Chapter 4, the Virtual Gyroscope directly replaces a conventional gyroscope in an attitude filter. With an additional attitude sensor in the loop, the sign ambiguity of the angular velocity estimate was resolved without requiring an estimate of angular acceleration and the performance of the simulated attitude filter is assessed.

Chapter 5 concludes this thesis with a recapitulation of the discoveries and new questions identified in this research.
Chapter 2

DERIVING THE VIRTUAL GYROSCOPE

2.1 Accelerometer and Virtual Gyroscope Measurement Models

The objective of this chapter is to present the mathematical formulation for the Virtual Gyroscope (VG) that inputs the acceleration vector measurements from an array of accelerometers and outputs an estimate of the magnitude of the angular velocity components. The goal is to derive a measurement equation of the form given by equation (2.1), which matches typical linear gyroscope models in the literature [7][18]. It will be shown in this chapter that it is unphysical to use this measurement model for the VG, which introduced the risk that existing filtering techniques which assume a gyroscope model like equation (2.1) will be less effective for the Virtual Gyroscope.

\[ \tilde{\omega} = |\omega| + \eta_\omega + \beta_\omega \]  

(2.1)

In equation (2.1), the VG’s estimate of the unsigned angular velocity was denoted \( \tilde{\omega} \) as a function of the true angular velocity vector \( \omega \) and the error processes \( \eta_\omega \) and \( \beta_\omega \). The usual notation for an estimate of a state is to use the symbol \( \hat{\text{ }\text{ } } \) over the variable, while the symbol \( \tilde{\text{ }\text{ } } \) is reserved for a measurement. In this case, however, the least squares estimator that was used in this section is playing the roll of a virtual sensor, so its output was considered a measurement \( \tilde{\omega} \), while the symbol \( \hat{\omega} \) was reserved for the filtered estimate of angular velocity discussed in Chapter 4. In equation (2.1), the error of the VG was divided into two parts: \( \eta_\omega \) is the error contribution that is caused by the zero mean white noise in the accelerometers, while \( \beta_\omega \) is the time-varying bias as a function of the bias of the individual accelerometer elements.
An additional objective of this chapter was to understand the statistics of these two random processes by analytically mapping of noise and bias of the individual accelerometer elements to noise and bias of the VG as a whole.

Consider an array of $N$ three-axis accelerometers distributed throughout a three-dimensional rigid body that is free to translate and rotate. An example of one possible array containing fourteen accelerometers for a 1U CubeSat is shown below in Figure 2.1.

![14 Accelerometer Array](image)

Figure 2.1: 14 Sensor Array Configuration
For the purpose of deriving a Virtual Gyroscope for nanosatellite applications, it makes sense to choose a small cube as the rigid body, but much of the following analysis is agnostic to the geometry of the object to which the sensors are mounted. This formulation is based on the derivation in [11], which uses the following reference frame assignment.

![Coordinate Frame](image)

**Figure 2.2: Coordinate Frame [11]**

In Figure 2.2, the body frame is shown with respect to the inertial frame. The position vectors of the center of mass of the vehicle and the location of the sensor can be expressed in either body or inertial coordinates. For the subsequent analysis, all calculations are done in the body frame of the vehicle, although the calculations can be performed in other reference frames with the appropriate transformation.
The acceleration vector measured by accelerometer \( i \) is modeled by the expression

\[
\ddot{a}_i = a_i + \eta_i + \beta_i, \tag{2.2}
\]

where \( a_i \) is the true acceleration experienced by sensor \( i \) in the body frame, \( \eta_i \) is a zero-mean Gaussian white noise process with variance \( \sigma_i^2 \) representing the sensor white noise, and \( \beta_i \) is the sensor’s bias with rate of change \( \dot{\beta}_i = \eta_{\beta,i} \) where \( \eta_{\beta,i} \) is Gaussian white noise with zero mean and variance \( \sigma_{\beta,i}^2 \). The simulation for the accelerometers in Chapter 3 that is used to test the VG will include a more sophisticated model including error sources like quantization and sensitivity to temperature, but the model proposed in equation (2.2) will be used for the subsequent analysis.

The true acceleration experienced by sensor \( i \) can be divided into two components: the acceleration due to the motion of the spacecraft’s center of mass with respect to the inertial frame, and the acceleration due to the rotation of the spacecraft’s body frame with respect to the inertial frame. The acceleration is a function of the angular acceleration experienced by the spacecraft \( \alpha \), the angular velocity of the spacecraft \( \omega \), and the combined center-of-mass (COM) acceleration of spacecraft \( \bar{a}_{COM} \), all with respect to the inertial frame expressed in body frame coordinates. Specifically,

\[
a_i = \bar{a}_{COM} + (\alpha^x + (\omega^x)^2)r_i, \tag{2.3}
\]

where \( v^x \) is the standard cross-product matrix defined in [8] satisfying the property \( v \times u = v^x u \) and \( r_i \) is the position vector of the \( i \)th accelerometer in the body frame. Because the calculations are done in the body frame here, the individual sensor measurements must be rotated from the sensor frame into the body frame before attempting to determine \( \dot{\omega} \) in the Virtual Gyroscope. Potential errors induced by uncertainty in alignment and position of the accelerometer are assumed to be
included in the noise and bias terms for the sensor as a whole. Under this assumption, there is no need to individually consider uncertainty in the position vector \( r_i \) and uncertainty in the rotation of \( \tilde{a}_i \) to the body frame from the sensor frame.

To derive an expression for unsigned angular velocity from the measured acceleration, equation (2.3) is substituted into equation (2.2):

\[
\tilde{a}_i = \bar{a}_{COM} + (\alpha^x + (\omega^x)^2)r_i + \eta_i + \beta_i. \tag{2.4}
\]

Because each term in equation (2.4) is a three-component vector, this is a system of three equations and six desired unknowns (\( \alpha \) and \( \omega \)), plus additional unknowns in the center of mass acceleration and error terms. Because the contribution of center-of-mass acceleration to the acceleration measurement is independent of the location of the sensor in the body frame, the \( \bar{a}_{COM} \) term can be eliminated by differencing the measurements of two sensors. Considering sensors \( i = 1 \) and \( 2 \), subtracting \( \tilde{a}_1 - \tilde{a}_2 \) yields the expression

\[
\Delta\tilde{a}_{1,2} = (\alpha^x + (\omega^x)^2)(r_1 - r_2) + (\eta_1 - \eta_2) + (\beta_1 - \beta_2) \tag{2.5}
\]

which is simplified to

\[
\Delta\tilde{a}_{1,2} = (\alpha^x + (\omega^x)^2)\Delta r_{1,2} + \Delta\eta_{1,2} + \Delta\beta_{1,2}, \tag{2.6}
\]

where \( \Delta r_{1,2} = r_1 - r_2 \), \( \Delta\eta_{1,2} = \eta_1 - \eta_2 \) and \( \Delta\beta_{1,2} = \beta_1 - \beta_2 \). Note the nonlinearity of equation (2.6). The square of the cross product matrix \( (\omega^x)^2 \) will be nonlinear in the components of angular velocity. This is the reason the VG’s measurement will be unsigned. If only accelerometers are used in the hypothetical spacecraft, this sign ambiguity is impossible to resolve unless an accurate signed estimate of angular acceleration is obtained (a case discussed in [11]). In practice, however, the estimate
of angular acceleration that the Virtual Gyroscope produces is too noisy to be much good. Instead, Chapter 4 will introduce an easy (but more computationally intensive) approach to resolve the sign ambiguity that does not require knowledge of the angular acceleration.

Ultimately, $\Delta \tilde{a}_{i,j}$ will be taken for many sensors and a least squares solution will be used to estimate $\tilde{w}$.

2.2 Solution without Noise or Bias

Before developing the least squares solution that includes sensor noise and bias, first consider an ideal case where the measured acceleration is free from noise and bias ($\tilde{a}_i = a_i$). Although unphysical, this simplified model will be instructive to develop the least squares model noisy case. Under this assumption, equation (2.6) becomes

$$\Delta \tilde{a}_{1,2} = (\alpha^x + (\omega^x)^2)\Delta r_{1,2}. \quad (2.7)$$

Letting the subscripts $x$, $y$, and $z$ denote the vector components in body coordinates, equation (2.7) is expanded into component form as

$$\Delta \tilde{a}_{1,2} = \begin{bmatrix} -\omega_y^2 - \omega_z^2 & \omega_x \omega_y - \alpha_z & \omega_x \omega_z + \alpha_y \\ \omega_x \omega_y + \alpha_z & -\omega_x^2 - \omega_z^2 & \omega_y \omega_z - \alpha_x \\ \omega_x \omega_z - \alpha_y & \omega_y \omega_z + \alpha_x & -\omega_x^2 - \omega_y^2 \end{bmatrix} \Delta r_{1,2}. \quad (2.8)$$

This is a system of three equations and six unknowns, but it is nonlinear in the components $\omega$. Instead, each nonlinear term will be defined as an independent unknown to produce a linear system of nine unknowns instead of a nonlinear system of six.
unknowns given by

\[ \Delta \tilde{a}_{1,2} = \begin{bmatrix} -x_2 - x_3 & x_4 - \alpha_z & x_5 + \alpha_y \\ x_4 + \alpha_z & -x_1 - x_3 & x_6 - \alpha_x \\ x_5 - \alpha_y & x_6 + \alpha_x & -x_1 - x_2 \end{bmatrix} \Delta r_{1,2} \] (2.9)

where \( x = \begin{bmatrix} \omega_x^2 & \omega_y^2 & \omega_z^2 & \omega_x \omega_y & \omega_y \omega_z \omega_z \end{bmatrix} \). There will, of course, be constraints between the elements of \( x \) because it is defined by only three unknown quantities.

To use the standard linear least squares solution as presented in [11], the system must be rewritten in the form \( z = Au \) where \( z \) is a vector of known quantities, \( A \) is a known matrix, and \( u \) is the vector of unknowns. Performing the matrix multiplication,

\[ \Delta \tilde{a}_{1,2} = \begin{bmatrix} -x_2 - x_3 & x_4 - \alpha_z & x_5 + \alpha_y \\ x_4 + \alpha_z & -x_1 - x_3 & x_6 - \alpha_x \\ x_5 - \alpha_y & x_6 + \alpha_x & -x_1 - x_2 \end{bmatrix} \begin{bmatrix} \Delta r_{1,2x} \\ \Delta r_{1,2y} \\ \Delta r_{1,2z} \end{bmatrix} \] (2.10)

\[ \begin{align*}
-\Delta r_{1,2x} x_2 - \Delta r_{1,2x} x_3 + \Delta r_{1,2y} x_4 - \Delta r_{1,2y} \alpha_z + \Delta r_{1,2z} x_5 + \Delta r_{1,2z} \alpha_y \\
\Delta r_{1,2x} x_4 + \Delta r_{1,2x} \alpha_z - \Delta r_{1,2y} x_1 - \Delta r_{1,2y} x_3 + \Delta r_{1,2z} x_6 - \Delta r_{1,2z} \alpha_x \\
\Delta r_{1,2x} x_5 - \Delta r_{1,2x} \alpha_y + \Delta r_{1,2y} x_6 + \Delta r_{1,2y} \alpha_x - \Delta r_{1,2z} x_1 - \Delta r_{1,2z} x_2
\end{align*} \] (2.11)

\[ = A_{1,2} u \] (2.12)

Where \( u = \begin{bmatrix} x^T & \alpha_x & \alpha_y & \alpha_z \end{bmatrix}^T \) and \( A_{1,2} \) is defined as

\[ \begin{bmatrix} 0 & -\Delta r_{1,2x} & -\Delta r_{1,2x} & \Delta r_{1,2y} & \Delta r_{1,2z} & 0 & 0 & \Delta r_{1,2z} & -\Delta r_{1,2y} \\ -\Delta r_{1,2y} & 0 & -\Delta r_{1,2y} & \Delta r_{1,2x} & 0 & \Delta r_{1,2z} & -\Delta r_{1,2z} & 0 & \Delta r_{1,2z} \\ -\Delta r_{1,2z} & -\Delta r_{1,2z} & 0 & 0 & \Delta r_{1,2x} & \Delta r_{1,2z} & \Delta r_{1,2y} & -\Delta r_{1,2x} & 0 \end{bmatrix} \] (2.13)
This matches the formulation presented by Hamley and Crassidis in [11]. Consider 4 sensors where the acceleration differences are $\Delta \tilde{a}_{1,2}$, $\Delta \tilde{a}_{2,3}$, and $\Delta \tilde{a}_{3,4}$. In theory, $\Delta \tilde{a}_{1,3}$ could be used instead to solve the equation (provided certain conditions on the positions of the accelerometers with respect to the instantaneous axis of rotation are met), $\Delta \tilde{a}_{3,4}$ is used instead to mirror the later formulation with many more sensors. Augmenting the system of equations with the additional sensor information, the system becomes

$$z = Au$$  \tag{2.14}$$

where $z =$

$$\begin{bmatrix}
\Delta \tilde{a}_{1,2x} & \Delta \tilde{a}_{1,2y} & \Delta \tilde{a}_{1,2z} & \Delta \tilde{a}_{2,3x} & \Delta \tilde{a}_{2,3y} & \Delta \tilde{a}_{2,3z} & \Delta \tilde{a}_{3,4x} & \Delta \tilde{a}_{3,4y} & \Delta \tilde{a}_{3,4z}
\end{bmatrix}^T \tag{2.15}$$

, $u =$

$$\begin{bmatrix}
x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & \alpha_x & \alpha_y & \alpha_z
\end{bmatrix}^T \tag{2.16}$$

and $A =$

$$\begin{bmatrix}
0 & -\Delta r_{1,2x} & -\Delta r_{1,2x} & \Delta r_{1,2y} & \Delta r_{1,2z} & 0 & 0 & \Delta r_{1,2x} & -\Delta r_{1,2y} \\
-\Delta r_{1,2y} & 0 & -\Delta r_{1,2y} & \Delta r_{1,2x} & 0 & \Delta r_{1,2z} & -\Delta r_{1,2y} & 0 & \Delta r_{1,2x} \\
-\Delta r_{1,2z} & -\Delta r_{1,2z} & 0 & 0 & \Delta r_{1,2x} & \Delta r_{1,2y} & \Delta r_{1,2y} & -\Delta r_{1,2x} & 0 \\
0 & -\Delta r_{2,3x} & -\Delta r_{2,3x} & \Delta r_{2,3y} & \Delta r_{2,3z} & 0 & 0 & \Delta r_{2,3z} & -\Delta r_{2,3y} \\
-\Delta r_{2,3y} & 0 & -\Delta r_{2,3y} & \Delta r_{2,3x} & 0 & \Delta r_{2,3z} & -\Delta r_{2,3y} & 0 & \Delta r_{2,3x} \\
-\Delta r_{2,3z} & -\Delta r_{2,3z} & 0 & 0 & \Delta r_{2,3x} & \Delta r_{2,3y} & \Delta r_{2,3y} & -\Delta r_{2,3x} & 0 \\
0 & -\Delta r_{3,4x} & -\Delta r_{3,4x} & \Delta r_{3,4y} & \Delta r_{3,4z} & 0 & 0 & \Delta r_{3,4z} & -\Delta r_{3,4y} \\
-\Delta r_{3,4y} & 0 & -\Delta r_{3,4y} & \Delta r_{3,4x} & 0 & \Delta r_{3,4z} & -\Delta r_{3,4y} & 0 & \Delta r_{3,4x} \\
-\Delta r_{3,4z} & -\Delta r_{3,4z} & 0 & 0 & \Delta r_{3,4x} & \Delta r_{3,4y} & \Delta r_{3,4y} & -\Delta r_{3,4x} & 0
\end{bmatrix} \tag{2.17}$$
The $A$ defined here is simply a block matrix constructed from $A_{1,2}$ defined in equation (2.13), as well as $A_{2,3}$ and $A_{3,4}$ defined similarly. This system can be solved for an estimate of $u$ (denoted $\hat{u}$) with the least squares solution $\hat{u} = (A^T A)^{-1} A^T z$.

The pseudoinverse matrix $(A^T A)^{-1} A^T$ will always exist for this system, provided the accelerometers do not exist in the same location of the spacecraft body. For any accelerometer array that is symmetrical with respect to the body axis, some $\Delta r$ elements in $A$ will be zero. Notice, however, that each row contains the differences in both $x$, $y$, and $z$ components so as long as the accelerometers are at different locations, at least two elements in each row will be nonzero.

In this case, where $\tilde{a}_i = a_i$, the estimate of the unknown state vector $\hat{u} = u$. Each product of angular velocity components in the vector $x$ will be exactly the product of the true angular velocity components, and the unsigned estimate $\tilde{\omega}$ is simply

$$\tilde{\omega} = \begin{bmatrix} \sqrt{x_1} & \sqrt{x_2} & \sqrt{x_3} \end{bmatrix}^T.$$  

(2.18)

The estimate of the signed angular velocity vector $\hat{\omega}$ will be equal to $\tilde{\omega}$ with one of the $3!$ (three-factorial) possible combinations of signs for the components. Alternatively, once one component magnitude has been found from the first three components of $x$ and a sign is chosen for it, the signs of the other two components could be identified using $x_4 = \omega_x \omega_y$, $x_5 = \omega_x \omega_z$, or $x_6 = \omega_y \omega_z$. Although this method would reduce the candidates for $\hat{\omega}$ to only two possible vectors, in practice the sensor noise will make it difficult to correctly identify the signs using $x_4$, $x_5$, or $x_6$, especially at low angular velocities where the signal to noise ratio for the accelerometers is low.
2.3 Solution with White Noise

In a more physical case where noise is present, it will be helpful to augment the $A$ and $z$ matrices with additional accelerometer measurements and position differences. Each set of three rows in $A$ and $z$ is the same set of equations with each sensor index incremented by one. This makes it relatively simple to build up the $A$ and $z$ matrix for an arbitrary number of sensors, as long as their positions in the body frame are known.

The problem of deriving angular velocity from multiple acceleration measurements was considered by Hamley and Crassidis in [11], although their analysis did not consider sensor bias, only zero-mean Gaussian white noise. Borrowing the results from [11], the matrices $A$ and $z$ can be built elegantly for an arbitrary number of sensors ($N$) with the following algorithm.

First, the matrix $L$ is defined by the expression

$$L = \left[ I_{(N-1)\times(N-1)} - U_{(N-1)\times(N-1)} \right] \odot I_{3\times3}$$

where $I$ is the identity matrix of the specified size, $U$ is an upper-shift binary matrix with ones in the superdiagonal and zeros everywhere else, and $g = \left[ 0 \ 0 \ \ldots \ -1 \right]^T$. $\odot$ is the Kronecker product. $L$ is referred to as the “fourth differencing matrix” because (when multiplied by an arbitrary column vector $v$ of length $3N$) it creates a column vector of differences between every fourth elements from that vector. For example, for $N = 3$, the size of $L$ will be a six by nine matrix. If this $L$ is multiplied by a column vector of length nine (for example containing the measurement vectors of three accelerometers), the output $Lv = [(v_1 - v_4)^T \ (v_2 - v_5)^T \ \ldots \ (v_6 - v_9)^T]^T$. 

16
Using the notation from [11], the least squares model becomes

\[
z = \begin{bmatrix} \Delta J & \Delta \mathcal{H} \end{bmatrix} u + \mathcal{W}, \tag{2.20}
\]

\[
\tilde{z} = \begin{bmatrix} \Delta J & \Delta \mathcal{H} \end{bmatrix} u. \tag{2.21}
\]

In equation (2.20), \( u \) is the vector of unknowns as defined previously, and \( z = La \) is the same \( z \) defined previously in the no noise case when \( a_i = \tilde{a}_i \). In the noisy case, however, only a measurement of \( \tilde{z} = z - \mathcal{W} = L\tilde{a} \) is accessible, where

\[
\tilde{a} = \begin{bmatrix} \tilde{a}_1^T & \tilde{a}_2^T & \ldots & \tilde{a}_N^T \end{bmatrix}^T
\]

(2.22)

is a column vector of acceleration measurements. The block matrices that make up the \( A \) matrix as defined in [11] are \( \Delta J = LJ \), \( \Delta \mathcal{H} = L\mathcal{H} \), and \( \mathcal{W} = L\eta \) where

\[
J = \begin{bmatrix} J_1^T & J_2^T & \ldots & J_N^T \end{bmatrix}^T,
\]

(2.23)

\[
\mathcal{H} = \begin{bmatrix} r_1^{xT} & r_2^{xT} & \ldots & r_N^{xT} \end{bmatrix}^T,
\]

(2.24)

\[
\eta = \begin{bmatrix} \eta_1^T & \eta_2^T & \ldots & \eta_N^T \end{bmatrix}^T,
\]

(2.25)

and

\[
J_i = \begin{bmatrix} 0 & -r_{xi} & -r_{xi} & r_{yi} & r_{zi} & 0 \\
-r_{yi} & 0 & -r_{yi} & r_{xi} & 0 & r_{zi} \\
-r_{zi} & -r_{zi} & 0 & 0 & r_{xi} & r_{yi} \end{bmatrix}
\]

(2.26)

Upon careful inspection, it can be seen that this nesting of equations and definitions will produce the same \( A = \begin{bmatrix} \Delta J & \Delta \mathcal{H} \end{bmatrix} \) matrix defined previously in the case where
$N = 4$ from equation (2.15). This approach, however, is easily scalable in code for arbitrary size and shape of accelerometer arrays.

The least squares solution is the same as defined previously except $\tilde{z}$ is used instead of the now inaccessible vector $z$. The estimate of $u$ becomes

$$\hat{u} = (A^T A)^{-1} A^T \tilde{z}. \quad (2.27)$$

### 2.4 Solution with White Noise and Bias

If white noise and bias are present, the measurement equation for the individual accelerometer is given by equation (2.2). The same least squares solution given in equation (2.27) can be used here with $\tilde{z} = L\tilde{a}$, but now $\tilde{a}$ includes both the noise and bias terms.

Consider a specific moment in time, when the sum of the random variables $\eta_i$ and $\beta_i$ crystalize to a specific number, which is the residual $e_i(t)$, such that the true acceleration at time $t$ is $a_i(t) = \tilde{a}_i(t) - e_i(t)$. To better understand the residual of the Virtual Gyroscope as whole, this section will analytically derive the residual of the Virtual Gyroscope as a function of the residual of the individual accelerometers.

The analysis here focuses on the $N = 6$ accelerometer array, in which the vectors and matrices are of reasonable length to be written down. This array configuration has one accelerometer on each face of a cube of side length $l$. At a specific instance in time, the measurement vector of the accelerometers is an eighteen element column
vector

\[ \tilde{a} = \begin{bmatrix} \tilde{a}_1 \\ \tilde{a}_2 \\ \vdots \\ \tilde{a}_6 \end{bmatrix} \]  

(2.28)

where each \( \tilde{a}_i \) is the three component measurement from accelerometer \( i \). Substituting in the formula for \( a_i \) using the residual (and dropping the \( t \) argument for this particular instance in time),

\[ \tilde{a} = \begin{bmatrix} \tilde{a}_1 \\ \tilde{a}_2 \\ \vdots \\ \tilde{a}_6 \end{bmatrix} = \begin{bmatrix} a_1 + e_1 \\ a_2 + e_2 \\ \vdots \\ a_6 + e_6 \end{bmatrix} \]  

(2.29)

Preparing the least squares solution from the previous question, the matrix \( L \) is given by a fifteen by eighteen matrix with ones on the main diagonal and a minus ones diagonal shifted right by three indices:

\[
L = \begin{bmatrix}
1 & 0 & 0 & -1 & 0 & 0 & 0 & \ldots \\
0 & 1 & 0 & 0 & -1 & 0 & 0 & \ldots \\
& & \ddots & & \ddots & & & \\
\ldots & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\
\ldots & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\
\end{bmatrix}.
\]  

(2.30)
When this matrix is applied to $\tilde{a}$, it will form a fifteen element column vector that is the difference in corresponding components of consecutive accelerometers. That is,

$$\tilde{z} = L\tilde{a},$$

$$= \begin{bmatrix}
(a_1 - a_2) + (e_1 - e_2) \\
(a_2 - a_3) + (e_2 - e_3) \\
\vdots \\
(a_5 - a_6) + (e_5 - e_6)
\end{bmatrix},$$

$$= z + \Delta e.$$  \hspace{1cm} (2.31

And $z$ is an eighteen element column vector of the difference in true accelerations for consecutive accelerometers, and $\Delta e$ is the corresponding difference in residuals.

For the $N = 6$ accelerometer array, the $A$ matrix will be fifteen by nine, and the estimate of the solution vector $\hat{u}$ will be

$$\hat{u} = (A^T A)^{-1} A^T \tilde{z}$$

$$= (A^T A)^{-1} A^T (z + \Delta e)$$

$$= u + (A^T A)^{-1} A^T \Delta e$$

$$= u + u_{error}.$$  \hspace{1cm} (2.34

So the residual $u_{error}$ will be the differences in the residuals of the individual accelerometers, scaled by a constant matrix. Expanding the first element of $\hat{u}$ to observe the estimate of the square of the first component of angular velocity ($\hat{w}_x^2$),

$$\hat{u}_1 = \hat{w}_x^2$$

$$= w_x^2 + u_{1, error}$$

20
where \( u_{1,\text{error}} \) is given by an equation involving the components of the individual residual vectors \( e_{ix}, e_{iy}, \) and \( e_{iz} \) given by

\[
\begin{align*}
\frac{29}{70l}e_{1x} - \frac{41}{70l}e_{2x} + \frac{1}{10l}e_{3x} + \frac{1}{10l}e_{4x} - \frac{1}{70l}e_{5x} - \frac{1}{70l}e_{6x} \\
+ \frac{1}{14l}e_{1y} + \frac{1}{14l}e_{2y} - \frac{1}{2l}e_{3y} + \frac{1}{2l}e_{4y} - \frac{1}{14l}e_{5y} - \frac{1}{14l}e_{6y} \\
- \frac{1}{70l}e_{1z} - \frac{1}{70l}e_{2z} + \frac{1}{10l}e_{3z} + \frac{1}{10l}e_{4z} - \frac{41}{70l}e_{5z} + \frac{29}{70l}e_{6z}
\end{align*}
\]

and the estimate of the unsigned x-component of angular velocity is then

\[
\tilde{\omega}_x = \sqrt{\omega^2_x + u_{1,\text{error}}}. \tag{2.41}
\]

Similar expressions can be written down for the y and z component, each becoming the square root of the true angular velocity plus a linear combination of the individual accelerometer residuals. The factor of \( \frac{1}{l} \) present in each term of equation (2.38) means the residual will decrease the further apart the accelerometers are spaced in the spacecraft. Recall one of the goals of this chapter is to find a measurement model in the form of equation (2.1). It is clear from equation (2.41) that it would be inaccurate to model \( \tilde{\omega} \) as a linear combination of the true value, the white noise, and the bias because the real equation for the measurement is nonlinear. Instead, a more accurate measurement equation for the \( j \)th components of \( \tilde{\omega} \) would be

\[
\tilde{\omega}_j = \sqrt{\omega^2_j + \eta_{\omega,j} + \beta_{\omega,j}}. \tag{2.42}
\]

In this formulation, the noise vectors \( \eta_\omega \) and \( \beta_\omega \) are Gaussian white noise with zero mean and integrated white noise with variance dependent on both the variance of the individual accelerometers noise power and the array configuration.
In equation (2.38), all residual terms are present, and the coefficients depend on the size and shape of the array. In the cross product terms $u_4 = \omega_x \omega_y$, $u_5 = \omega_x \omega_z$, and $u_6 = \omega_y \omega_z$, the residual of the other component is not present in the expansion. This means that $u_4$ is independent of the $z$-component of the residuals, $u_5$ is independent of the $y$-component, and $u_6$ is independent of the $x$-component. The acceleration residuals follow another trend. The residual of $\alpha_x$ only depends on the $y$ and $z$ residuals, and so on for $\alpha_y$ and $\alpha_z$.

### 2.5 Chapter 2 Conclusion

The standard bias prediction schemes for attitude filters like in [7] take advantage of the fact that the residual of the angular velocity estimate is the sum of white noise and bias where the rate of change of the bias is white noise. As seen in equation (2.42), this is not the case for the Virtual Gyroscope because of the square root, so standard bias prediction schemes are expected to be less effective here. The realization that the error terms for the Virtual Gyroscope are more complicated than that of a standard gyroscope introduces a new objective for this thesis: can existing attitude filtering strategies using a linear measurement model be used for the Virtual Gyroscope? To begin to answer this question, a multiplicative extended Kalman filter (MEKF) is applied to estimate attitude from the VG and a quaternion sensor in Chapter 4. The MEKF is implemented using the standard gyroscope model given by equation (2.1) instead of equation (2.42) to evaluate the efficacy of the classic gyroscope noise model to filter the VG.
This chapter describes the simulation environment and mathematical models used to test the Virtual Gyroscope (VG). It details the dynamics model used as the simulation plant, the model of the accelerometer sensors that provide the VG inputs, and the techniques used to evaluate the VG’s performance characteristics in a way analogous to real gyroscopes.

### 3.1 Simplified Satellite Dynamics Model

The spacecraft is modeled as a rigid body that is free to translate and rotate in a zero potential environment. The rigid body is acted upon by a fictitious force $F_c$ and torque $T_c$, which are applied to the center of mass to produce a translation acceleration $\bar{a}_{CoM}$ and a rotation acceleration $\alpha$. The center of mass acceleration $\bar{a}_{CoM}$ of the body due to the force, is experienced equally at every point on the rigid body.

The state of the dynamic system is described by the set of four vectors: the CoM position $r$, the velocity $v$, the attitude quaternion $q$, and the angular velocity vector $\omega$, which obey the dynamics equations

\[
\bar{a}_{CoM} = \frac{F_c}{m},
\]

\[
\alpha = I^{-1}(T_c - w \times Iw),
\]
and kinematic equations

\[
\begin{bmatrix}
\dot{\mathbf{r}} \\
\dot{\mathbf{v}}
\end{bmatrix} =
\begin{bmatrix}
\mathbf{v} \\
\mathbf{a}_{CoM}
\end{bmatrix},
\tag{3.3}
\]

\[
\dot{\mathbf{q}} =
\begin{bmatrix}
\dot{\epsilon} \\
\dot{q}_4
\end{bmatrix} =
\begin{bmatrix}
\frac{1}{2}(q_4\mathbb{I} + \epsilon^\times)\omega \\
-\frac{1}{2}\epsilon^T\omega
\end{bmatrix},
\tag{3.4}
\]

and

\[
\dot{\omega} = \alpha,
\tag{3.5}
\]

where \(m\) is the mass of the rigid body, \(I\) is the inertia matrix, \(\mathbb{I}\) is the identity matrix, and \(\epsilon\) and \(q_4\) are the vector and scalar parts, respectively, of the rigid body’s attitude quaternion. The fourth dimensional quaternion representation was used for attitude representation to avoid rotational singularity inherent to Euler angle or direction cosine matrix kinematics formulations.

All vectors are expressed in a right-handed body fixed coordinate system at the center of mass of the rigid body, and an accelerometer is affixed to the rigid body at position \(r_i\). The inertial acceleration \(a_i\) experienced by the sensor is a combination of the rotational and linear states of the system from equations (3.1) through (3.5) such that

\[
a_i = a_{CoM} + (\alpha^\times + \omega^\times\omega^\times)r_i,
\tag{3.6}
\]

where the second term is the sum of the centripetal and tangential accelerations experienced by the sensor due to the rotation.

To simulate the rigid body dynamics with ideal acceleration measurements at specific points in the body in MATLAB and Simulink, the initial position, velocity, attitude quaternion, and angular velocity were defined, along with the mass and inertia matrix of the rigid body. From there, equations (3.1) through (3.5) were numerically
integrated with a fixed simulation frequency of 2000 Hz to find the state of the system at future times for a choice of $F_c$ and $T_c$, which are defined as either time varying signals or constants for different test cases. The frequency of 2000 Hz was chosen to assure that the system was simulated at a much higher frequency than the dynamics of the system, to mitigate errors in state propagation. From the propagated system states, equation (3.6) was used to determine the acceleration experienced by an array of sensors at various positions in the rigid body. This acceleration $a_i$ was taken to be the “true” acceleration of the system and was the input of the accelerometer model to determine the measurement of acceleration $\hat{a}_i$ from each sensor.

### 3.2 Accelerometer Model

There are many proposed methods to model accelerometers available in the literature, each with different levels of complexity and fidelity to reality based on the model’s intended use cases. In order to test the Virtual Gyroscope with comparable characteristics to commercially available MEMS sensors, the model must be consistent with the parameters from a manufacturer’s datasheet. Furthermore, if an accelerometer is modeled using the characteristics from the manufacturer’s datasheets, the sensor simulation must reproduce the advertised performance metrics.

The structure of the accelerometer model was created based on the documentation for the the imuSensor class in Mathwork’s Navigation Toolbox [4]. The mathematics used to emulate the sensor’s measurement process are from Applied Mathematics in Integrated Navigation Systems, by Rogers [18]. The simulation was validated by successful reproduction of the most dominant stochastic error characteristics of a real accelerometer.
3.2.1 Accelerometer Error Sources

The measurement error sources for an accelerometer can be broken into two categories: deterministic and stochastic error sources. Deterministic errors are error sources that can be modeled by algebraic or dynamic equations based on the physics of the system. If all the relevant states of the sensor are known, the exact value of the contribution of these error sources can, in principle, be calculated and removed from the measurement. Stochastic error sources on the other hand are modeled as random processes agnostic to the underlying physics of the system. Although the physics of the sensor does, of course, give rise to these sources of error, the physics is either too complicated or insufficiently understood to model deterministically. The final model chosen for the accelerometer is constructed in Simulink as shown in Figure 3.1, where each subsystem box is labeled with its purpose. The contents of each subsystem and function block are covered in more detail below; Figure 3.1 is only intended to provide a high level overview for the model architecture.

Figure 3.1: MEMS Accelerometer Sensor Model
3.2.1.1 Stochastic Error Sources

The stochastic error sources are all characterized by the white-noise spectrum on the accelerometer output. The value of the noise at each time step was computed in simulation from a difference equation, which corresponds to passing unity white noise through the appropriate shaping filter for each process, as described below. The difference equations correspond to rational transfer functions with integer exponents, which limits the type of random process that can be simulated in this way.

White Noise/Velocity Random Walk:

This noise process was modeled in the White Noise Drift subsystem in Figure 3.1. White noise has constant power spectral density given by equation (3.7),

\[ S(f) = Q^2, \]  

(3.7)

where \( S(f) \) is the power spectral density of the noise process at a frequency \( f \), and \( Q \) is the white noise coefficient [10]. The value of \( Q \) can be extracted directly from the experimentally determined Root Allan Variance (RAV) curve which is present in many manufacturers’ datasheets in units of acceleration per square root of frequency. The mathematics used to compute the RAV curve are shown in more detail in Appendix A, and covered in detail by [10] and other sources. The process used to extract the white noise coefficient from a RAV curve is discussed in more detail in section 3.2.2. On the specifications table of most datasheets, this coefficient is labeled as “Noise Density”. \( Q \) may also be given as velocity random walk in dimensions of velocity per square root of time, which describes the effect of integration of the random acceleration over time when the acceleration measurement is used to estimate velocity.
The value of $Q$ is independent of the sampling frequency of the accelerometer, but the actual variance of the resulting noise depends on the sampling frequency. Because the white noise has the same powers at all frequency, the effective standard deviation of the white noise contribution is $Q$ multiplied by the square root of the sampling frequency to reflect the same total noise power in the discrete-time case that can be found in the continuous-time case. The white noise term at each measurement time step can be calculated by

$$\beta_{WN}(k) = w(k)Q\sqrt{f_s},$$

(3.8)

where $w(k)$ is an instance of a normally distributed random variable with unity variance and zero mean at discrete time step $k$.

Acceleration Random Walk:

This noise process was modeled in the Random Walk Drift subsystem in Figure 3.1. Acceleration random walk has power spectral density

$$S(f) = \left( \frac{K}{2\pi} \right)^2 \frac{1}{f^2},$$

(3.9)

where $K$ is the acceleration random walk coefficient in units of acceleration times square root of frequency [10]. Unlike $Q$, the acceleration random walk coefficient is not readily available on many accelerometer datasheets’ specification tables, likely because it’s contribution is small for accelerometers over short measurement periods.

Like white noise bias, the acceleration random walk must also be adjusted for the operating frequency of the sensor. The resulting difference equation for the acceleration random walk bias is given by

$$\beta_{ARW}(k) = \beta_{ARW}(k - 1) + w(k)K\sqrt{f_s}T_s.$$  

(3.10)
This process is an approximation integration in discrete time of a random variable with standard deviation $K\sqrt{f_s}$, where $T_s$ is the sampling step size. $T_s = 1/f_s$, so equation (3.10) can be rewritten as

$$\beta_{ARW}(k) = \beta_{ARW}(k-1) + \frac{w(k)K}{\sqrt{f_s}}.$$  

(3.11)

Bias Instability:

Despite its presence in many accelerometer models in the literature, this noise process was not included in the accelerometer model for the Virtual Gyroscope, shown in Figure 3.1. Bias instability is intended to model the change in the bias over long time periods. The power spectral density for bias instability is given by the equation

$$S(f) = \begin{cases} \left(\frac{B^2}{2\pi}\right)^{\frac{1}{2}}, & \text{if } f \leq f_0, \\ 0, & \text{if } f > f_0, \end{cases}$$

(3.12)

where $B$ is the bias instability coefficient and $f_0$ is the cut-off frequency [10]. Although it is possible to estimate the bias instability coefficient from a RAV curve for a physical sensor, it is not possible to exactly simulate a random process with power spectral density inversely proportional to frequency like equation (3.12) in a discrete time simulation that uses only linear difference equations, and usually computationally complex fractional calculus based solutions must be implemented [12]. There are multiple methods proposed in the literature to approximate the $1/f$ noise with linear difference equations [4], represented as Gauss Markov processes [18][10], or more complicated algorithms [16][9], but each of these methods is either beyond the scope of the VG simulation or cause the value of $Q$ and $K$ to fail to match experimentally determined RAV curves from real sensors. To best match the results presented in various manufacturer datasheets, bias instability error term was not included in the
sensor model. Although this concession may cause the long-term bias drift to poorly represent the bias instability of a real sensor, the error characteristics are largely unaffected over short time scales. On longer time scales, a Kalman filter that was paired with the Virtual Gyroscope accounts for small errors in the sensor model.

3.2.1.2 Deterministic Error Sources

Although the deterministic error sources included in the model do follow exact equations with no random variables, it is important to recognize that the parameters used in these equations such as initial offset, resolution, and temperature scale factor vary from sensor to sensor and only the distribution of possible values may be known. In the simulation of the accelerometer sensors using this model, the value of the parameters that are used in the equations must be chosen randomly from within a known probability distribution at the beginning of the simulation (and held constant throughout). Some parameters can be calibrated for, such as constant bias and axis misalignment. In this case, the measurement can be corrected in software before being used in the Virtual Gyroscope algorithm.

Constant Errors:

- Axis misalignment should be constant for a given sensor mounted to a rigid spacecraft, so its effect is assumed to be calibrated for and ignored.

- Constant bias ($\beta_{\text{offset}}$) describes the constant difference between the ground truth acceleration and the acceleration measurement, before any other dynamic bias terms are added. This term (often referred to as “0g Offset” in datasheets) is selected from a small distribution of possible values from the sensor’s datasheet, when available.
Environmental Drift:

In the Environment Drift subsystem of Figure 3.1, the environmental bias $\Delta_e$ was calculated by

$$\Delta_e = (T_i - 25)T_{Bias}, \quad (3.13)$$

where $T_i$ is the temperature of the sensor in degrees Celsius and $T_{Bias}$ is the bias temperature coefficient. Temperature bias is sometimes also referred to as temperature slope or temperature rate. The bias temperature coefficient has units of $(\text{m/s}^2)/{^\circ\text{C}}$ and is assumed to be the same for each axis for the accelerometer sensor. Additionally, the sensor is assumed to be operated within the manufacturer-specified temperature range where this bias process is linear in temperature and equation (3.13) is valid. If the value of temperature bias is known exactly, the contribution of the environmental bias term $\Delta_e$ can be calculated from the temperature of the accelerometer. The value of the bias temperature coefficient will not be known exactly, but will fall within some narrow distribution of possible values. The expected value of this bias can then be subtracted from the sensor’s signal before using it in the Virtual Gyroscope algorithm, thus mitigating this term’s contribution to the angular velocity estimate. In practice, this assumption will require pairing each accelerometer with a temperature sensor (MEMS accelerometers usually have integrated temperature sensors), as well as determining the expected value of the bias temperature coefficient for each sensor before flight. For the Virtual Gyroscope, no attempt was made to mitigate environmental drift before inputting the signal to the algorithm. As will be shown in the subsequent sections, this contribution makes up only a small component of the total bias of the array based on the initial offset and acceleration random walk in the sensor array. Chapter 2 showed unmodeled bias (and resolution of sign ambiguity) of the VG will necessitate a bias prediction scheme for the VG as a whole anyway. Although it may be possible to implement a more complex filter that attempts to
predict the bias of the accelerometer individually, there are filters readily available in
the literature to predict gyroscope bias. In Chapter 4, one such filter is applied to
the Virtual Gyroscope to determine if the investigation of a more complex scheme is
warranted.

Temperature Scale Factor:

In Scale Factor Error Model subsystem in Figure 3.1, the signal is multiplied by a
temperature dependent scaling factor $s_T$, such that

$$s_T = 1 + \left( \frac{T_i - 25}{100} \right) T_{SF}, \quad (3.14)$$

where $T_{SF}$, called temperature scale factor in the model, is a multiplicative factor
that scales up or down the magnitude of the signal based on the temperature above
or below the reference temperature of 25 degrees Celsius. The distribution of possible
values for this term can be identified in most datasheets.

Quantization Error:

In the quantization stage of the model, shown in Figure 3.1, the algorithm checks
if the signal is within the measurement range and rounds the value to the nearest
quantized signal value based on the sensor’s digital resolution, in the case of digital
sensors, or based on the Analog to Digital Converter (ADC) of the satellite’s hardware
digital resolution in the case of analog sensors. Sensor resolution depends on the mea-
asurement range of the sensor for digital accelerometers with selectable measurement
ranges and is usually specified in least significant bit (LSB) per unit acceleration. This
quantization function is denoted $f_q(x_s)$ where $x_s$ is the signal without quantization.
3.2.1.3 Final Mathematical Model

Combining all the noise and bias contributions yields the final mathematical model represented by Figure 3.1.

\[ \tilde{a}_i = f_q(s_T[a_i + \beta_{\text{offset}} + \Delta e + \beta_{\text{WN}} + \beta_{\text{ARW}}]) \] (3.15)

3.2.2 Stochastic Noise Model Validation: ADXL 357

To validate the accelerometer noise model relative to the stochastic noise sources, the ADXL 357 accelerometer from Analog Devices was simulated. In a successful validation, an RAV curve from the simulation yielded similar parameters to the RAV curve in the datasheet from which the input parameters were extracted. First, the relevant accelerometer parameters were extracted from a RAV curve and the specifications table of the datasheet [1].

To find the value of \( Q \) from a RAV curve, one has to fit a line with a slope of \(-1/2\) to the downward sloped portion of the RAV curve. The value of \( Q \) can be taken directly from the equation for the best fit line at a value of \( \tau = 1 \) [10]. To obtain \( K \), one has to fit a line with a slope of \(+1/2\) to the positive sloped portion a RAV curve. The value of the acceleration random walk will be the value of the linear fit at \( \tau = 3 \) [10]. Mathwork’s \texttt{allanvar} function was used to create the RAV curve for the simulated sensor [3].

Analog Devices presents RAV curves for each axis. Because the simulated error model calculates error components independent of axis, it is sufficient for model validation to choose any one axis from the ADXL 357 to reproduce the results. If the accelerometer was modeled for a real mission using physical sensors, it would be necessary to
test each axis separately because the performance of each axis is not always identical. Figure 3.2 shows the RAV curve for the x-axis of the ADXL 357. The x-axis was chosen because it has the most clearly visible $+1/2$ slope which is necessary to determine the acceleration random walk coefficient in equation (3.11).

Each color line in Figure 3.2 corresponds to a different static test run with integration times between 0.01 and 1000 seconds. However due to the truncation of the horizontal axis, only the yellow curve reaches a slope of $+1/2$ by an integration time of 1000 seconds. Bringing the data from this curve into MATLAB and fitting the $\pm 1/2$ lines to the curve, the coefficients are found to be $Q = 4.9 \times 10^{-4}$ m/(s$^2\sqrt{\text{Hz}}$) and $K = 6.0 \times 10^{-6}$ m/s$^2\sqrt{\text{Hz}}$. 

![Figure 3.2: ADXL 357 X Axis Root Allan Variance Curves [1].](image)
To emulate a controlled testing environment, the temperature was set to 25°C which eliminates environmental drift and scale factor error. Any constant bias was assumed to be calibrated for and removed. The ADXL 357 advertises a resolution of $1/52000 \text{ g/LSB}$, which corresponds to a simulation resolution of roughly $2 \times 10^{-4} \text{ m/(s}^2\text{LSB)}$.

The various parameters and coefficients shown in Table 3.1 were then plugged into the accelerometer model, which was run with zero input for 10 million seconds at a sampling frequency of 20 Hz. The resulting RAV curve was obtained from the simulated data for this validation is shown in Figure 3.4 below. The noise coefficients extracted from the datasheet matched the noise coefficients from the simulation within 1%.
With the stochastic noise model validated for the ADXL 357 accelerometer, the next step was to conduct the Root Allan Variance test for the whole Virtual Gyroscope using simulated ADXL 357 accelerometers forming a spatially distributed sensor array.

3.3 Array Configurations

It was expected that the performance of the VG would be greatly dependent on the number of accelerometers in the array. More accelerometers means more data for the least squares fit, but there also is a limit to how many accelerometers can be realistically used in a CubeSat, especially the 1U model. For this research, a generous upper bound of fifty-four accelerometers was chosen for the largest array. A diagram
of this array is shown in Figure 3.5. The blue dots represent sensor locations, while the red planes are the faces of the CubeSat.

![54 Accelerometer Array (Orthographic)](image)

**Figure 3.5: 54 Accelerometer Array**

The sensors are placed on the outer face of the spacecraft to increase the distance from the sensor to the axis of rotation. At an angular velocity of 1 rad/s, the maximum acceleration in the array body that an accelerometer could measure would be 0.2 m/s$^2$, if the sensor is at the maximum distance from the axis of rotation. This is a difficult signal to work with, especially when the initial offset can vary on the order of 0.25 m/s$^2$, so it is important to use every strategy possible to increase the signal.
to noise ratio. The array sizes that were tested use fifty-four, twenty-eight, fourteen (this will be the “nominal” array configuration to which others are compared), eight, and six sensors. The six sensor array places one sensor in each face, the eight sensor places one sensor on each corner, the fourteen sensor array combines the sensors of the six and eight arrays. The fourteen sensor array places one sensor in each of the locations as the fourteen sensor array, then duplicates the amount of sensors each with a slight inward shift along the same radius. An orthographic view of each array is shown in Figure 3.6.

![Various Array Configurations](image)

**Figure 3.6: Various Array Configurations**

MEMS accelerometers are very small, and it is reasonable to assume many sensors can fit in the spacecraft’s body. For a 1U CubeSats, twenty-eight to fifty-four sensors
may be unfeasible based on the size limitations of the spacecraft. The nominal array of fourteen sensors is very reasonable for a 1U CubeSat, especially when sampled at the chosen sample rate of 20 Hz.

In the MATLAB code, these arrays are stored as $3 \times N$ matrices of position vectors in body coordinates named $R_6$, $R_8$, $R_{14}$, etcetera. The sensors fill the array from alternating sides of the spacecraft body, so that the sensor pairs formed by the Virtual Gyroscope algorithm have the largest average distance between them.

In the VG algorithm, it is important to pair each accelerometer with a sensor on the far side of the spacecraft body. If the accelerometers are paired with a directly adjacent sensor, the signal to noise ratio will be too small to be detected through the noise. Of course, which sensor pairs yield the best signal to noise ratio changes depending on the instantaneous axis of rotation. In principle an adaptive sensor pairing method could be designed to populate the $R_N$ matrices at each time step based on magnitude of the measured signal. Although an interesting concept, it is unlikely this would significantly improve the performance, especially considering the unknown bias in the accelerometers.

3.4 Testing the Virtual Gyroscope

The characterization of the Virtual Gyroscope began with analysis of the gyroscope alone, without any filters or sensor fusion. Many test cases were considered, each designed to isolate the effect a particular factor had on the performance of the VG. The factors considered were accelerometer signal to noise ratio, accelerometer quantity and quality, and temperature variation.
The nominal simulation parameters are used as the default values for parameters in the Virtual Gyroscope algorithm. Unless otherwise specified in a particular test case, the number used for a particular parameter or variable will be the value shown in Table 3.1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>geometry</td>
<td>cube</td>
<td>n/a</td>
</tr>
<tr>
<td>side length</td>
<td>10</td>
<td>cm</td>
</tr>
<tr>
<td>mass</td>
<td>2</td>
<td>kg</td>
</tr>
<tr>
<td>simulation frequency</td>
<td>2000</td>
<td>Hz</td>
</tr>
<tr>
<td>measurement frequency</td>
<td>20</td>
<td>Hz</td>
</tr>
<tr>
<td>K</td>
<td>$6.0 \times 10^{-6}$</td>
<td>m/s²/√Hz</td>
</tr>
<tr>
<td>Q</td>
<td>$4.9 \times 10^{-4}$</td>
<td>m/(s²/√Hz)</td>
</tr>
<tr>
<td>Resolution</td>
<td>$2 \times 10^{-4}$</td>
<td>m/(s²/LSB)</td>
</tr>
<tr>
<td>$\beta_{Offset}$</td>
<td>$\pm 125$</td>
<td>mg</td>
</tr>
<tr>
<td>$T_{Bias}$</td>
<td>$\pm 0.2$</td>
<td>(m/s²)/°C</td>
</tr>
<tr>
<td>$T_i$</td>
<td>25</td>
<td>°C</td>
</tr>
<tr>
<td>$T_{SF}$</td>
<td>$\pm 0.01$</td>
<td>%/°C</td>
</tr>
<tr>
<td>Array</td>
<td>$R_{14}$</td>
<td>n/a</td>
</tr>
</tbody>
</table>

In Table 3.1, $\beta_{Offset}$, $T_{SF}$, and $T_{Bias}$ are specified by the bounds from the ADXL 357 datasheet from which the actual value used in the simulation will be randomly selected for each accelerometer. These ranges appear to be three standard deviation bounds based on histograms in the datasheet [1]. The mass and geometry choices are consistent with the form factor of a 1U CubeSat. The simulation frequency and measurement frequency are chosen sufficiently high to reduce numerical errors in the underlying dynamics simulation and sampled rapidly to avoid aliasing. $T_i$ is the standard temperature where there will be no temperature scale factor or environmental drift.
3.4.1 Performance Metrics

Before introducing the various test cases, the metrics used to analyze the performance of the Virtual Gyroscope must be defined.

For conventional MEMS gyroscopes, the Root Allan Variance is a typical method for evaluating sensor performance. The procedure for measuring the noise coefficients from an RAV curve is almost identical to the procedure used for an accelerometer, except the coefficient $K$ is rate random walk instead of acceleration random walk and has units of $\text{rad/s}\sqrt{\text{Hz}}$ and the white noise coefficient $Q$ has units of $\text{rad}/(\text{s}\sqrt{\text{Hz}})$. In Chapter 2 it was predicted that the error of the Virtual Gyroscope cannot cleanly be divided into the sum of a zero mean Gaussian white noise process and an integrated Gaussian white noise process. Despite this, Root Allan Variance can still be used to characterize the relative performance changes of the Virtual Gyroscope in various test cases. The coefficients $Q$ and $K$ are thought of as measuring the effective white noise and rate random walk coefficients that the Virtual Gyroscope if the measurement equation was (unphysically) modeled as $\tilde{\omega} = \omega + \eta_\omega + \beta_\omega$. These coefficients will also be used in Chapter 4 to define the covariance matrices for the angular velocity measurement used in the attitude filter.

It only makes sense to use Root Allan Variance in situations where the true angular velocity experienced by the gyroscope is known so the error in the measurement is the input to the RAV algorithm (see Appendix A). This is why Root Allan Variance tests typically use a stationary sensor, when the true angular rates can be estimated accurately. Depending on the sensitivity of the gyroscope, it is sometimes necessary to account for the rotation of the Earth during the data collection period. The VG simulation, however, has access to both the estimated state and the true angular velocity. As such, RAV can be used in every constant temperature test case, as long
as the residual is calculated and input into the algorithm instead directly inputting the measurement vector. Application of RAV curves was still limited to constant temperature test cases to better isolate the bias drift due to the acceleration random walk of the accelerometers.

In addition to the RAV curves, it was instructive to plot the residual of the unsigned angular velocity estimate, both over time and as a probability density function. The graphs and statistics of the residual were useful as more qualitative metrics of VG performance, but it is important to recognize that the residual will not have constant mean and variance because of the acceleration random walk in the array’s members. The standard deviation and mean of the residual are not perfect metrics for analyzing array performance because of this limitation, but they still provided an effective at-a-glance method to gauge relative performance between simulation test cases, especially when the residual was averaged across many trials with different noise seeds.

Test cases A, B, and C attempted to analyze the noise properties of the Virtual Gyroscope with no initial accelerometer bias and constant temperature (and thus no environmental drift or scaling). These test cases attempted to isolate the effect array size and accelerometer noise has on the overall white noise and random walk of the Virtual Gyroscope. They also reveal how the signal to noise ratio for the whole Virtual Gyroscope is heavily dependent on the signal to noise ratio of the individual accelerometers. In test case A, the Virtual Gyroscope was analyzed with zero input corresponding to a stationary object, and the residual error is analyzed via Allan Variance. In test case B, the Virtual Gyroscope is simulated in the scenario of a constant angular velocity rotation. Next, test Case C analyzed the Virtual Gyroscope with variable angular velocity due to a sinusoidal control torque.

The next two tests motivated sensor fusion for bias prediction, as detailed in Chapter 4. In test case D and E, each sensor had a temperature slope, scale factor, and
initial offset chosen from the datasheet’s suggested distribution, and the effect of the
individual array element bias was examined. Test case D allowed the temperature to
vary sinusoidally, thus causing environmental drift in each sensor. Finally, test case
E added the initial offset of each sensor, further obfuscating the real angular velocity
with error. As seen in test case E, bias prediction is surely necessary to predict the
angular velocity accurately, although this was already a requirement for the resolution
of sign ambiguity in the angular velocity estimate.

Each test case constituted eight trials with different noise seeds and the five previously
defined array sizes. The array performance was also be examined with more and less
accurate sensors, by scaling the noise coefficient used by the VG. The results of each
run will be compared by analyzing the residual unsigned angular velocity, which is
defined as the difference between the true unsigned angular velocity and estimate of
the angular velocity from the Virtual Gyro.

3.4.2 Test Case A: Zero Input Without Initial Bias

This test case assumed the Virtual Gyroscope experiences no external forces and
torques at a constant standard temperature. This corresponds with a more idealized
version of the physical test conditions often used for Allan Variance tests, in which
the Gyroscope is set on a stationary level surface at as constant a temperature as
possible. Should the VG really be set on a flat surface, of course, the accelerometers
would sense acceleration due to gravity $g$ away from the contact surface. This test case
was more analogous to an Allan Variance test of the Virtual Gyroscope in free fall,
or (more realistically), the Virtual Gyroscope with the acceleration due to gravity
subtracted from each sensor reading. The initial conditions and noise parameters
(from the ADXL 357) for the physics simulation are shown in Table 3.2. The initial
bias was set to zero here, and the temperature is 25 $^\circ$C. First, eight simulations
were conducted each array, for a total of 40 runs. The fourteen sensor array was also tested twice more, with the input noise coefficient increased by a factor of ten and decreased by a factor of ten to test the effect of different accelerometer qualities. For each run, the Root Allan Variance was calculated, then a white noise coefficient and rate random walk coefficient for the VG is determined.

<table>
<thead>
<tr>
<th>Table 3.2: Test Case A: Initial Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paramater</td>
</tr>
<tr>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Position</td>
</tr>
<tr>
<td>Velocity</td>
</tr>
<tr>
<td>Angular Velocity</td>
</tr>
<tr>
<td>Quaternion</td>
</tr>
<tr>
<td>Initial Sensor Bias</td>
</tr>
</tbody>
</table>

Before conducting Root Allan Variance tests for the 40 runs, consider the output of the Virtual Gyroscope for the first 1000 seconds in Figure 3.7. Only the x-axis is shown, but due to the symmetry of the sensor array, the behavior of each axis is statistically identical although each axis has an independent noise seed. This assumption is not valid for the real ADXL 357, which has different characteristics for each axis [1].
The estimate was not centered on the true value because the VG is estimating the square of the angular velocity; the Virtual Gyroscope will return a positive angular velocity, even if the true value is negative. For the given input, the average of the estimate was 0.091 rad/s (5 deg/s) above the true value.

Plotting a histogram of the residual, it was clear the error is also not normally distributed. The square of the residual was a half Gaussian, which was indicative of the white noise contribution to the accelerometer errors. In 1000 seconds, the acceleration random walk does not contribute excessively to the measurement, so the bias in the accelerometers was still low for this trial.
Because the true angular velocity was zero in test case A, the accelerometers are only producing noise without any nonzero true signal. When the accelerometer readings were subtracted from each other in the VG, the noise power was doubled, but the underlying distribution remained unchanged. Multiplying the resulting random vector with the least squares pseudoinverse matrix resulted in a linear combination of multiple independent noise processes, which will also yield a Gaussian distribution because the linear combinations of two Gaussian white noise processes is also Gaussian white noise. Only when the square root of the state vector was taken to calculate the individual angular velocity components is the Gaussian distribution concealed.

The characteristics of the error were evaluated further using the Allan Variance. In order to capture the effect of the acceleration random walk in the sensors, each run
must last at least a hundred thousand seconds, containing two million data points. This large data requirement made running the Virtual Gyroscope fairly slow in MATLAB, although certain design considerations such as precomputing matrix inverses sped up the process. For later tests that do not use Allan Variance, fewer data points were required.

The Root Allan Variance plots for the 14 sensor array are shown below. The 24 blue lines are the RAV curves for the x, y, and z axis of the 8 trials.

![Test Case A: N = 14 Root Allan Variance](image)

**Figure 3.9: Example Root Allan Variance Curve for Test Case A with 14 Sensors**

As with the experimental RAV curves from the accelerometer datasheet, the curves agree closely across various runs for the white noise section (fit with a red line), but tended to diverge for the rate random walk section (fit with a green line). This is
reflected in the much larger deviation in rate random walk coefficient values, while the white noise coefficients have a tighter spread. Note, for some of the linear fits, the RAV never actually made it to a slope of a half, so the linear fits are not always tangent to the RAV curve. This means the error for the value of the rate random walk coefficient $K$ was large, and the reported value were overestimates. The lack of clean $+1/2$ slopes indicated that the Virtual Gyroscope did a good job at attenuating rate random walk, likely because the individual ADXL 357 accelerometers exhibit little acceleration random walk. This does not mean bias drift can be ignored, however, because the initial bias offsets and temperature drift has yet to be included in the model. See test cases D and E for more details on these factors.

For the other array sizes, the plots are not shown here, although it was necessary to compute them to extract the noise characteristics of the VG. Instead, the resulting noise characteristics for each run are summarized in Table 3.3 along with the 1σ errors.

<table>
<thead>
<tr>
<th>Trial</th>
<th>$Q \text{ rad/s}/\sqrt{\text{Hz}}$</th>
<th>$K \text{ rad/s}/\sqrt{\text{Hz}}$</th>
<th>Mean Error rad/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_6$</td>
<td>$1.3 \times 10^{-2} \pm 3.1 \times 10^{-4}$</td>
<td>$7.9 \times 10^{-5} \pm 3.0 \times 10^{-5}$</td>
<td>0.14</td>
</tr>
<tr>
<td>$R_8$</td>
<td>$9.8 \times 10^{-3} \pm 4.0 \times 10^{-4}$</td>
<td>$7.1 \times 10^{-5} \pm 3.1 \times 10^{-5}$</td>
<td>0.11</td>
</tr>
<tr>
<td>$R_{14}$</td>
<td>$9.2 \times 10^{-3} \pm 2.5 \times 10^{-4}$</td>
<td>$6.1 \times 10^{-5} \pm 2.5 \times 10^{-5}$</td>
<td>0.10</td>
</tr>
<tr>
<td>$R_{28}$</td>
<td>$8.0 \times 10^{-3} \pm 2.6 \times 10^{-4}$</td>
<td>$5.2 \times 10^{-5} \pm 2.2 \times 10^{-5}$</td>
<td>0.087</td>
</tr>
<tr>
<td>$R_{54}$</td>
<td>$8.1 \times 10^{-3} \pm 2.7 \times 10^{-4}$</td>
<td>$4.96 \times 10^{-5} \pm 2.3 \times 10^{-5}$</td>
<td>0.088</td>
</tr>
<tr>
<td>10x Noise</td>
<td>$2.9 \times 10^{-2} \pm 7.8 \times 10^{-4}$</td>
<td>$1.9 \times 10^{-4} \pm 7.9 \times 10^{-5}$</td>
<td>0.32</td>
</tr>
<tr>
<td>0.1x Noise</td>
<td>$2.9 \times 10^{-3} \pm 8.5 \times 10^{-5}$</td>
<td>$1.9 \times 10^{-5} \pm 7.7 \times 10^{-6}$</td>
<td>0.032</td>
</tr>
</tbody>
</table>

The improvement with array size is plotted in Figure 3.10.
All considered noise metrics decreased fairly rapidly for array sizes between 6 and 14, but for larger array sizes an increase in array size yielded diminishing returns. Based on this curve, the 14 sensor array was chosen as the nominal array configuration for future tests.

In the last two columns of Table 3.3, it was clear that improving sensor quality does decrease all three noise metrics, but not by the same factor that is applied to the accelerometer noise coefficients. If the accelerometer noise coefficients are changed by a factor of ten, the Virtual Gyroscope noise coefficients only change by a factor of about 3.1. Only when the noise coefficients are changed by a factor of a hundred do the noise characteristics change by a full order of magnitude.
3.4.3 Test Case B: Constant Nonzero Input Without Initial Bias

In this test case the Virtual Gyroscope was rotated at constant angular velocity, increasing the signal to noise ratio of the individual sensor elements. The main comparison will be the residual angular velocity magnitude.

The initial conditions for test case B are shown in the table 3.4 below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acceleration Random Walk Coefficient</td>
<td>$K$</td>
<td>$6.0 \times 10^{-6}$</td>
<td>m/s$^2\sqrt{\text{Hz}}$</td>
</tr>
<tr>
<td>White Noise Coefficient</td>
<td>$Q$</td>
<td>$4.9 \times 10^{-4}$</td>
<td>m/s$^2/\sqrt{\text{Hz}}$</td>
</tr>
<tr>
<td>Position</td>
<td>$r_0$</td>
<td>[0;0;0]</td>
<td>m</td>
</tr>
<tr>
<td>Velocity</td>
<td>$v_0$</td>
<td>[0;0;0]</td>
<td>m/s</td>
</tr>
<tr>
<td>Angular Velocity</td>
<td>$\omega_0$</td>
<td>Variable</td>
<td>rad/s</td>
</tr>
<tr>
<td>Quaternion</td>
<td>$q_0$</td>
<td>[0;0;0;1]</td>
<td>n/a</td>
</tr>
<tr>
<td>Initial Sensor Bias</td>
<td>$B_0$</td>
<td>0</td>
<td>m/s$^2$</td>
</tr>
<tr>
<td>Temperature</td>
<td>$T_0$</td>
<td>25</td>
<td>°C</td>
</tr>
</tbody>
</table>

The constant angular velocity was varied, and the residual angular velocity magnitude was used to compare the results. Figure 3.11 plots the average and variance of the residual angular velocity magnitude at various angular rates.
Notice that the residual mean and variance decrease for non-zero angular velocity because the signal to noise ratio of the accelerometers is increased. The error actually approaches zero for larger angular velocities, but keep in mind that this case does not include initial offset in the bias. It may be possible to use the Virtual Gyroscope in other applications where the bias can be filtered out, but it is unlikely a CubeSat would require accurate attitude knowledge while rotating at 10 rad/s or greater. It is important to recognize that the residual angular velocity magnitude is an inferior metric for Virtual Gyroscope performance than Allan Variance. Because of the acceleration random walk (and offset and environmental drift in later test cases), the residual is not expected to zero mean. Furthermore, the mean of the residual will be time dependent as bias drift causes the estimate to “wander” above and below
the true value. This test case merely attempted to point out that VG noise power decreases as expected as the signal to noise ratio of the individual sensors increases.

3.4.4 Test Case C: Variable Angular Velocity Without Initial Bias

The Virtual Gyroscope also performed better if it was accelerating. Recall, the equation for the acceleration of an individual sensor is given by (3.6).

This test case observed the influence of the $\alpha$ term on the Virtual Gyroscope performance. Like case B, this section was not intended to provide a precise mathematical description of the Virtual Gyroscope’s performance, but instead to illustrate another factor that affects the VG performance.

In this test case, only a single run was conducted with a sinusoidal control torque (in Newtons) given by the equation

$$T_c = (1 \times 10^{-5}) \sin(2\pi t/1000) \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$  \hspace{1cm} (3.16)

The other initial conditions are the same as test case A, as shown in table 3.2. The relevant states of the first 4000 seconds of the simulation are shown in figure 3.12.
In 3.12, the spread of the residual changed dynamically with the angular acceleration and velocity, reaching a minimum as acceleration peaks.

3.4.5 Test Case A, B, and C Conclusions

Based on tests cases A, B, and C alone, it was concluded that in the absence of temperature fluctuation and initial bias, the Virtual Gyroscope measures the magnitude of the angular rate well for higher rotational rates and poorer for low rates. The quality of the estimate did improve with increased array size, but the improvement levels off rapidly beyond 14 sensors. For spacecraft applications of the Virtual Gyroscope, this is insufficient. There are few CubeSat applications that require accurate angular rate knowledge when the spacecraft is rotating very rapidly. Most pointing and tracking applications on the CubeSat scale require higher accuracy at low angular rates than the Virtual Gyroscope is providing.
3.4.6 Test Case D: Variable Temperature

This test repeated the simulation from case C, except this time the temperature was varied sinusoidally between the upper and lower bound of the allowable range from the ADXL 357 datasheet. According the datasheet, the ADXL 357 offset versus temperature ranges between ±0.2mg/°C and the sensitivity change from temperature was between ±0.01%/°C [1]. From the histogram of possible values provided by the manufacturer, this appears to be a 3σ range.

The value of these parameters for each sensor was chosen at random from a zero mean Gaussian distribution. The temperature is varied between -40 and 125 °C with a period of 4000 seconds.

The simulation results are presented in figure 3.13.

![Figure 3.13: Case D: Variable Temperature Simulation](image)

In the middle plot on figure 3.13, it can be seen that the environmental drift of the individual sensors led to a cumulative environmental drift for the entire Virtual
Gyroscope. At high temperatures, the VG underestimated the true angular velocity magnitude, while the trend reverses at low temperatures. At standard temperature, the results matched test case C. It is a good sign that the drift of the bias with temperature was small, even though all 14 sensors have different temperature slopes. The fact that this bias was small means it can be accounted for more readily with bias prediction in Chapter 4.

3.4.7 Test Case E: Initial Bias

Finally, the initial offset of the individual sensors was considered. In this simulation, every noise source accounted for in the Virtual Gyroscope simulation was active. The individual sensors had initial bias, separate temperature slopes and scale factors, white noise, and acceleration random walk. The temperature, angular velocity, and control torque all vary with time.

The values of the initial sensor bias were chosen randomly from the ranges of initial offset presented in the ADXL 357 datasheet [1]. This initial offset has a $3\sigma$ range of $\pm 25mg$.

Running the simulation with the initial offset of all 14 sensors yields the states shown in figure 3.14.
In Figure 3.14, the true unsigned angular velocity is shown in black and the measurement from the Virtual Gyroscope is shown in red. There are a few interesting things to note here. First, the individual initial offset of the accelerometers translated to a new initial offset for the array as a whole. Unfortunately, when the value of the angular velocity changes, this offset did not manifest as a simple vertical shift to the bias of the angular velocity. As seen in the second curve in figure 3.14, there was a scaling effect to the true signal when this offset was applied. Even though the true signals all have the same value (because the torque is applied to each axis), the different initial offsets lead to different measurements for the three components.
3.5 Chapter 3 Conclusions

This chapter introduced the simulation methodology for the spacecraft model and the accelerometer array. A model of the ADXL 357 accelerometer was created and tested using Root Allan Variance. The Virtual Gyroscope was then tested. Many of the main results supported and added qualifications to the conclusions in Chapter 2. The VG improved with more sensors and higher quality sensors, although the improvements yielded diminishing returns after fourteen sensors. The VG measured more accurately at higher angular velocities, while the low signal to noise ratio obscured the acceleration for low rotation rates and angular accelerations.

Sensor fusion will be necessary to get useful performance out of the Virtual Gyroscope for CubeSat applications. Although the unfiltered performance may be sufficient for measuring rapid rotation rates after calibrating away the initial offset, an attitude sensor will be necessary both to resolve sign ambiguity and filter out the bias.
In this chapter, the Virtual Gyroscope replaced a standard MEMS gyroscope in a Multiplicative Extended Kalman Filter (MEKF) attitude estimation scheme for a 1U CubeSat. Chapter 2 and 3 demonstrated that the Virtual Gyroscope has a time-varying bias that is proportional to the square root of a linear combination of measurement errors of the accelerometers that make up the array. The square root limits the Virtual Gyroscope to only estimating unsigned angular velocity, and the unknown biases of the individual array elements yield large errors in the unsigned estimate. An attempt was made to resolve these limitations by sensor fusion between the Virtual Gyroscope’s unsigned estimate of angular velocity and a measurement of the attitude. It was found that if the attitude measurements are made available to the filter at the same sample rate as the Virtual Gyroscope, the properly tuned filter would rely completely on the attitude measurements with no improvement in quality based on the Virtual Gyroscope, even with noise added to the attitude sensor. In fact, the attitude filter obtained the same quality of estimate with the Virtual Gyroscope as when the angular velocity measurement was not used at all with the rapidly sampled attitude measurement. When the covariance matrices of the filter were tuned to force the filter to rely more on the VG measurements, the quality of the attitude estimate rapidly degraded. An additional test case considered attitude measurements at a much lower frequency than the sample rate of the Virtual Gyroscope and found the attitude filter still delivered a higher quality estimate when tuned to ignore the Virtual Gyroscope. As a whole, this chapter demonstrates that the current formulation of the Virtual
Gyroscope algorithm is a poor choice for attitude determination in a CubeSat. The following sections detail the methodology and tests that led to this conclusion.

4.1 Attitude Sensor Measurement Model

The simulation from Chapter 3 used the simplified spacecraft dynamics model to provide the inputs to the accelerometer array model, which in turn provided the inputs to the Virtual Gyroscope algorithm. In this chapter, the simulation was necessarily augmented with an additional sensor to measure the attitude of the spacecraft. There are many different ways to estimate a nanosatellite’s attitude, but the most common methods over recent launches is to use magnetometers and sun sensors [13]. These methods tend to provide poorer performance than a star tracker, but in exchange offer a less expensive attitude determination solution. Star trackers, sun sensors, and magnetometers all function as attitude sensors by collecting multiple body-frame measurements of vectors with known components in the inertial frame (such as the direction of the sun, stars, or the local magnetic field). The method of determining attitude from many vector measurements is known as Wahba’s problem, and multiple solutions are available in the literature [7][14]. In short, the measured vector in the body frame is compared to the expected value in inertial coordinates, and an algorithm like TRIAD (for two vector measurements) or more commonly QUEST is used to solve for the attitude [14]. Attitude sensor modeling was not the focus of this research, so it was left to the imagination where the vector measurements provided by the attitude sensor come from.

It was critical for this research that the attitude estimate from the measurement vector was not too good. If the attitude from the sensor model was free from noise, there would be little need for sensor fusion with a gyroscope (Virtual or otherwise).
and it will be difficult to assess the VG’s performance in the context of attitude estimation. In the following tests, this was shown to be the case even with a noisy sensor if the attitude is sampled as rapidly as the VG. In the hope of better studying the Virtual Gyroscope (and to more realistically approximate the sensor’s available for 1U CubeSats), noise was added to the vector measurements from the fictitious attitude sensor such that the unfiltered attitude estimate has roughly a $5^\circ$ $3\sigma$ error. This error corresponds with fairly low accuracy sensors according to NASA’s 2022 State-of-the-Art for Small Spacecraft Technology [5]. As was found in this chapter, even with strong additive noise to the attitude measurement, the Virtual Gyroscope’s rate estimate failed to improve the filter’s performance.

It was assumed that three vector measurements ($\tilde{b}_1$, $\tilde{b}_2$, and $\tilde{b}_3$) were available of known physical vectors ($p_1$, $p_2$ and $p_3$) in the inertial frame. To generate these measurements in simulation, Gaussian white noise $\eta_q$ of variance $\sigma_q^2$ was added to the true quaternion from the dynamics simulation to get three noisy quaternions. For this simulation, the chosen value for the measurement noise standard deviation is $\sigma_q = .015$ (dimensionless because it was applied to the quaternion), which yields a $3\sigma$ accuracy of $5.2^\circ$ in the resulting angle measurements. That is, $\tilde{q}_i = q_i + \eta_{q,i}$ for $i = 1, 2, 3$ where $\tilde{q}_i$ is the $i$th measurement of the true quaternion $q$. These quaternion measurements were then brute-force re-normalized by dividing the quaternion vector by its own magnitude.

The measurement vector at time step $k$ for the attitude filter then became

$$\tilde{y}_k = \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \tilde{b}_3 \end{bmatrix} = \begin{bmatrix} A(\tilde{q}_1)p_1 \\ A(\tilde{q}_1)p_2 \\ A(\tilde{q}_3)p_3 \end{bmatrix},$$

(4.1)

where $A(\tilde{q}_i)$ is the attitude matrix corresponding with the $i$th quaternion measurement.
4.2 Extended Kalman Filter Formulation

The equations of quaternion kinematics are nonlinear, so an Extended Kalman Filter (EKF) must be used to linearize the equations of motion about the previous best estimate of the state at each time step. The Kalman Filter model used here is discussed at length in [7]. This section summarizes the relevant details of the model and applies it to the Virtual Gyroscope simulation.

The Kalman Filter from [7] was used “out of the box” without any modification to the gyroscope error model. Instead of using the estimate of angular velocity \( \hat{\omega} \) in the state vector, this filter instead uses the error in the gyro bias estimate \( \Delta \beta = \beta - \hat{\beta} \) where \( \beta \) is the true gyro bias such that

\[
\omega = \tilde{\omega} - \beta - \eta_u.
\] (4.2)

This was the model that was desired in Chapter 2, but was shown to not accurately model the VG error. The process noise for this model is defined by error \( \eta_u \) and the bias \( \beta \) with rate of change \( \dot{\beta} = \eta_v \). \( \eta_u \) and \( \eta_v \) are zero mean Gaussian white noise processes with variance \( \sigma_u^2 \) and \( \sigma_v^2 \), respectively. Looking back to Chapter 3, the standard deviations \( \sigma_u \) and \( \sigma_v \) should be taken to be the effective white noise coefficient \( Q \) and rate random walk coefficient \( K \), respectively, from a Root Allan Variance curve for the VG. These parameters will appear in the filter’s process noise covariance matrix and can be tuned to cause the filter to be more or less “trusting” of the measurements from the gyro with respect to the attitude sensor.
4.2.1 Multiplicative Quaternion Error

Along with the error in the bias estimate, the state vector for the Kalman Filter also includes quaternion error. Unlike typical vector errors, quaternions represent rotations, so it is physically meaningless to define the quaternion error as the true quaternion minus the estimate. Instead, the quaternion error $\delta q$ is defined as the overall rotation resulting from the rotation corresponding to the true quaternion followed by the inverse rotation represented by the estimated quaternion

$$\delta q = q \otimes \hat{q}^{-1}$$ (4.3)

where $\otimes$ is the quaternion multiplication operator defined by

$$q_A \otimes q_B = \begin{bmatrix} \Xi(q_B) & q_B \end{bmatrix} q_A$$ (4.4)

and $\hat{q}^{-1}$ is the quaternion inverse of the estimated quaternion. In equation (4.4) the matrix $\Xi(q)$ is defined as

$$\Xi(q) = \begin{bmatrix} q_4 I_3 + \epsilon^x \\ -\epsilon^T \end{bmatrix}$$ (4.5)

where $\epsilon$ is the quaternion vector and $q_4$ is the scalar component. The inverse quaternion is defined as

$$q^{-1} = \begin{bmatrix} -\epsilon \\ q_4 \end{bmatrix}$$ (4.6)

such that $q \otimes q^{-1} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$. 

62
Rather than using the whole error quaternion, the number of states is reduced to six by defining the state vector for the Kalman filter as

\[
\Delta x = \begin{bmatrix}
\delta \alpha \\
\Delta \beta
\end{bmatrix}
\] (4.7)

where \(\delta \alpha\) is the roll-pitch-yaw angle error which can be approximated as twice the vector component of the error quaternion \((\delta \varepsilon)\). This simplification is valid for small angle errors when the estimated quaternion is close to the true quaternion [7].

With the state vector process noises defined, the six-state Multiplicative Extended Kalman Filter can be implemented with the following algorithm:

### 4.2.1.1 Step 1: Initialize

- Initialize the estimate of the attitude quaternion \(\hat{q}_0\), the estimate of the bias \(\hat{\beta}_0\), and the error covariance matrix of the state vector \(P_0\).

For the simulation, these quantities are initialized somewhat arbitrarily to

\[
\hat{q}_0 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T
\] (4.8)

\[
\beta_0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}^T
\] (4.9)

\[
P_0 = I_6
\] (4.10)

As will be seen in the test cases, the filter quickly corrects these initial conditions, especially if \(P_0\) is not small. These initial conditions define the first values of the a priori estimates \(P_k^-, \hat{q}_k^-, \) and \(\hat{\beta}_k^-\) respectively where the superscript \(-\) denotes the estimate from the previous time step.
4.2.1.2 Step 2: Compute Kalman Gain

- First, compute the sensitivity matrix as a function of the a priori state.

\[
H_k(\hat{x}_k^-) = \begin{bmatrix}
(A(\hat{q}_k^-)p_1) \times & 0_{3 \times 3} \\
(A(\hat{q}_k^-)p_2) \times & 0_{3 \times 3} \\
(A(\hat{q}_k^-)p_3) \times & 0_{3 \times 3}
\end{bmatrix}
\]  
(4.11)

- In this simulation, the three known physical vectors are chosen to be arbitrary (different) unit vectors expressed in inertial coordinates. It is critical that the \( p_i \) vectors used in (4.11) are the same as the vectors used to compute the measurement in equation (4.1).

- Compute the Kalman Gain with the equation

\[
K_k = P_k^{-1}H_k^T(\hat{x}_k^-)[H_k(\hat{x}_k^-)P_k^{-1}H_k^T(\hat{x}_k^-) + R]^{-1}
\]  
(4.12)

where \( R = \sigma_R^2 I_9 \) is the measurement error covariance matrix.

4.2.1.3 Step 3: Update Using the New Measurements

- Define the predicted measurement from the a priori states

\[
h_k(\hat{x}_k^-) = \begin{bmatrix}
A(\hat{q}^-)p_1 \\
A(\hat{q}^-)p_2 \\
A(\hat{q}^-)p_3
\end{bmatrix}
\]  
(4.13)

- Update the state vector

\[
\Delta \hat{x}_k^+ = K_k[\hat{y}_k - h_k(\hat{x}_k^-)]
\]  
(4.14)
where the superscript + denotes the updated measurement.

- Unpack the updated angle and bias error from the state vector using the definition

\[
\Delta \hat{x}_k^+ \equiv \begin{bmatrix} \delta \hat{\alpha}_k^+ \\ \Delta \hat{\beta}_k^+ \end{bmatrix}
\]  
(4.15)

- Update and re-normalize the quaternion

\[
\hat{q}_k^+ = \hat{q}_k^- + \frac{1}{2} \Xi(\hat{q}_k^-) \delta \hat{\alpha}_k^+
\]  
(4.16)

\[
\frac{\hat{q}_k^+}{||\hat{q}_k^+||} \rightarrow \hat{q}_k^+
\]  
(4.17)

Here, \( a \rightarrow b \) means replace \( b \) with \( a \).

- Update the bias

\[
\hat{\beta}_k^+ = \hat{\beta}_k^- + \Delta \hat{\beta}_k^+
\]  
(4.18)

### 4.2.1.4 Step 4: Propagate in Discrete Time

- Estimate the angular velocity using the measurement directly from the best signed measurement from the VG and the bias estimate

\[
\hat{\omega}_k^+ = \tilde{\omega}_k - \hat{\beta}_k^+. 
\]  
(4.19)

See Section 4.2.2 for the procedure to choose the best sign for the measurement.

- Prepare the a priori estimate of the quaternion at the next time step:

\[
\hat{q}_{k+1}^- = \Omega(\hat{\omega}_k^+)\hat{q}_k^+
\]  
(4.20)
where

\[
\hat{\Omega}(\hat{\omega}_k^+) = \begin{bmatrix}
\cos \left( \frac{1}{2}||\hat{\omega}_k^+||\Delta t \right) I_3 - \psi_k^+ \\
-\psi_k^+ T & \cos \left( \frac{1}{2}||\hat{\omega}_k^+||\Delta t \right)
\end{bmatrix},
\]

(4.21)

and \( \Delta t \) is the measurement time step.

- Prepare the a priori estimate of the error covariance matrix:

\[
P_{k+1}^- = \Phi_k(\hat{\omega}_k^+) P_k^+ \Phi_k(\hat{\omega}_k^+)^T + \gamma_k Q_k \gamma_k^T,
\]

(4.23)

where

\[
Q_k = \begin{bmatrix}
(\sigma_v^2 \Delta t + \frac{1}{6} \sigma_u^2 \Delta t^3) I_3 & (\frac{1}{2} \sigma_u^2 \Delta t^2) I_3 \\
(\frac{1}{2} \sigma_u^2 \Delta t^2) I_3 & (\sigma_u^2 \Delta t) I_3
\end{bmatrix}.
\]

(4.24)

Additionally,

\[
\gamma_k = \begin{bmatrix}
-I_3 & 0_{3 \times 3} \\
0_{3 \times 3} & I_3
\end{bmatrix}
\]

(4.25)

and for some angular velocity \( \omega \)

\[
\Phi_k(\omega) = \begin{bmatrix}
\Phi_{11} & \Phi_{12} \\
\Phi_{21} & \Phi_{22}
\end{bmatrix},
\]

(4.26)

where the block matrices \( \Phi_{ij} \) are given by

\[
\Phi_{11} = I_3 - \omega \times \frac{\sin (||\omega||\Delta t)}{||\omega||} + \omega \times \omega \times \frac{(1 - \cos (||\omega||\Delta t))}{||\omega||^2},
\]

(4.27)

\[
\Phi_{12} = \omega \times \frac{(1 - \cos (||\omega||\Delta t))}{||\omega||^2} - I_3 \Delta t - \omega \times \omega \times \frac{||\omega||\Delta t - \sin (||\omega||\Delta t))}{||\omega||^3},
\]

(4.28)

\[
\Phi_{21} = 0_{3 \times 3},
\]

(4.29)
and

\[ \Phi_{22} = I_3. \]  

(4.30)

- Finally, the a priori bias for the next time step is simply given by

\[ \hat{\beta}^-_{k+1} = \hat{\beta}^+_{k}. \]  

(4.31)

If the attitude measurements are available at a greater time interval than the sample time step, the update step is only performed each time a new measurement is available instead of every time step. For more details on the derivation, see [7].

4.2.2 Resolving Sign Ambiguity

In theory, step 4 requires that the angular velocity update uses a signed angular velocity measurement for best performance. Without sensor fusion, the resolution of the sign ambiguity was a hard problem to solve because it required an accurate measurement of the angular acceleration of the spacecraft [11]. In the Kalman filter, however, the sign ambiguity problem was resolved by generating all eight sign combinations for the three angular velocity components and checking before the propagation step which sign combination was closest to the updated angular velocity from the previous time step. The set of signs that yielded the minimum difference between \( \tilde{\omega}_k \) and \( \tilde{\omega}_{k-1} \) was found, and that combination was used. Experimentally, this method was expected to work well as long as the frequency of the spacecraft’s rotation is much lower than the measurement frequency, but this was a requirement anyway to prevent aliasing. This assumption was not too restrictive because there are very few CubeSat operations that require a 1U spacecraft to rotate anywhere near twenty revolutions per second while requiring accurate attitude knowledge.
As will be seen in the test cases, the efficacy of this method was difficult to gauge because the filter relied on the propagated bias to update the angular velocity based on the attitude sensor without reliance on the VG measurement.

### 4.2.3 Measurement and Process Noise Covariance

The measurement noise covariance is given by

\[
R = \sigma_R^2 \mathbb{I},
\]

(4.32)

where \( \sigma_R^2 \) is the variance of the measurement of each component \( \tilde{b}_i \). For the chosen value of \( \sigma_q \) used to add noise to the quaternion measurement, the variance of the measurement error is experimentally determined to be about \( \sigma_R = 0.026 \) radians. This number was measured by generating many \( \tilde{b} \) and \( b \) vectors, then finding the variance of the residual, which is consistent with the measurement process model given by equation 4.1.

Normally, the Kalman filter is “tuned” by varying the values of the \( Q_k \) with respect to \( R \) by changing \( \sigma_u, \sigma_v, \) and \( \sigma_R \) until the filter performs well. The values chosen are \( \sigma_u = 9.2 \times 10^{-3} \) and \( \sigma_v = 6.1 \times 10^{-5} \), which are the effective \( Q \) and \( K \) coefficients, respectively, found for the Virtual Gyroscope in chapter 3 for the fourteen sensor array. The best performance occurred with \( \sigma \) fixed at the experimentally determined value. Although this selection of standard deviations did yield the best attitude estimate, it did so by causing the filter to rely almost solely on the attitude measurement with very little reliance on the angular velocity. It was found in the following test cases that the filter could be forced to rely more on the gyroscope by increasing \( \sigma_R \) (or decreasing \( \sigma_u \) and \( \sigma_v \)), but doing so resulted in poorer overall performance of the attitude filter.
Before presenting the testing, the factors used to compare filter performance are defined.

### 4.2.4 Performance Metrics

There are several main metrics for evaluating the performance of the attitude filter. To demonstrate how each metric was measured, the Virtual Gyroscope was tested in the nominal array configuration described in Table 4.1. Parameters with value “n/a” do not affect array performance when temperature is held constant at 25 °C.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation Frequency</td>
<td>2000</td>
<td>Hz</td>
</tr>
<tr>
<td>Gyro Measurement Frequency</td>
<td>20</td>
<td>Hz</td>
</tr>
<tr>
<td>Attitude Measurement Frequency</td>
<td>20</td>
<td>Hz</td>
</tr>
<tr>
<td>Geometry</td>
<td>cube</td>
<td></td>
</tr>
<tr>
<td>Side Length</td>
<td>10</td>
<td>cm</td>
</tr>
<tr>
<td>Mass</td>
<td>2</td>
<td>kg</td>
</tr>
<tr>
<td>Initial $\omega$</td>
<td>$[-0.5 \ 0 \ 0.5]^T$</td>
<td>rad/s</td>
</tr>
<tr>
<td>Initial $q$</td>
<td>$[0 \ 0 \ 0 \ 1]^T$</td>
<td></td>
</tr>
<tr>
<td>Array Configuration</td>
<td>$R_{14}$</td>
<td></td>
</tr>
<tr>
<td>Accelerometer $K$</td>
<td>$6.0 \times 10^{-6}$</td>
<td>m/s$^2\sqrt{\text{Hz}}$</td>
</tr>
<tr>
<td>Accelerometer $Q$</td>
<td>$4.9 \times 10^{-4}$</td>
<td>m/(s$^2\sqrt{\text{Hz}}$)</td>
</tr>
<tr>
<td>$T_{Bias}$</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>$T_{SF}$</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>$T_i$</td>
<td>25</td>
<td>°C</td>
</tr>
<tr>
<td>$\beta_{Offset}$</td>
<td>±25</td>
<td>mg</td>
</tr>
<tr>
<td>$\sigma_R$</td>
<td>0.026</td>
<td>rad</td>
</tr>
<tr>
<td>$\sigma_u$</td>
<td>$9.2 \times 10^{-3}$</td>
<td>rad</td>
</tr>
<tr>
<td>$\sigma_v$</td>
<td>$6.1 \times 10^{-5}$</td>
<td>rad/s</td>
</tr>
</tbody>
</table>
The simulation was run for 1000 seconds with a control torque of

\[ T_c = (1 \times 10^{-5}) \sin \left( \frac{2\pi t}{1000} \right) \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}. \]  \hspace{1cm} (4.33)

First, consider the evolution of the quaternion over time for the first twenty seconds.

![Figure 4.1: Filtered Quaternion for the Nominal Array Configuration](image)

In Figure 4.1, the filtered estimate shown in the solid blue line closely tracks the true quaternion in dashed red. The black dots show the noisy quaternion used in the attitude sensor measurement model. The filter appears to successfully track the true measurement. Even when given a bad initial guess for a starting point, the filter quickly corrects toward the true signal within about three seconds, as shown in Figure 4.2.
If the filter is converging, then the error should approach the identity quaternion. Calculating the error quaternion for both the noisy quaternion and the filtered results demonstrates that this was the case.
More useful is the angle error, which can be approximated as twice the vector component of the quaternion error [7]. Calculating this quantity and converting to degrees produced Figure 4.4.

![Figure 4.4: Filtered Angle Error](image)

In Figure 4.4, the angle error from the attitude sensors is plotted in black with the angle error from the filter plotted in red. Without the filter, the angle error is zero mean with a $3\sigma$ error of around $5^\circ$, as planned. With the filter, the angle error remains zero mean with a $3\sigma$ error of $1.07^\circ$. This shows a distinct improvement in pointing accuracy when the Extended Kalman Filter is employed. It did not, however, indicate that the Virtual Gyroscope was improving filter performance.

The estimate of the angular velocity was also greatly improved, both in accuracy and through resolution of the sign ambiguity. This was accomplished by the angular velocity update step in equation (4.19), where the estimated bias was added to the angular velocity measurement.
In Figure 4.5, the error in the angular velocity estimate before and after filtering is presented. Before filtering, the residual of the VG drifted on the order of 1.5 rad/s. With the filter, the residual of the estimated angular velocity had a $3\sigma$ of about 1.4 $\degree$/s. It was also observed that the quality of the filtered estimate did decrease for low angular rates as the noise in the Virtual Gyroscope increased, but not by a considerable amount.

For the subsequent tests, the main performance metrics used to evaluate the performance of the Virtual Gyroscope was the $3\sigma$ bounds on error angle and angular velocity error.
4.3 Testing the Attitude Filter with Rapidly Available Attitude Measurements

4.3.1 Accelerometer Quantity

At this point in the research, it was not yet apparent that the Virtual Gyroscope could not significantly affect the attitude estimate with the chosen process noise and measurement noise covariance matrices. The simulation was repeated with the $R_8$ through $R_{54}$ arrays. Table 4.2 shows the resulting $3\sigma$ error of runs for 100,000 seconds each. The long time span was chosen to improve the consistency of the standard deviation measurements. Three runs were conducted with each test case.

<table>
<thead>
<tr>
<th>Array</th>
<th>$3\sigma$ Error Angle (deg)</th>
<th>$3\sigma$ Rate Error (deg/s)</th>
<th>Max Angle Error (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R6</td>
<td>1.12</td>
<td>2.0</td>
<td>1.81</td>
</tr>
<tr>
<td>R8</td>
<td>1.13</td>
<td>1.73</td>
<td>1.77</td>
</tr>
<tr>
<td>R14</td>
<td>1.08</td>
<td>1.54</td>
<td>1.8</td>
</tr>
<tr>
<td>R28</td>
<td>1.11</td>
<td>1.63</td>
<td>1.75</td>
</tr>
<tr>
<td>R54</td>
<td>1.15</td>
<td>1.64</td>
<td>1.8</td>
</tr>
</tbody>
</table>

There was not a significant increase in performance as array size increased. This result was the first experimental sign that the filter was not benefiting from the VG.

4.3.2 Accelerometer Quality

The simulation was run two more times for 10,000 seconds using the $R_{14}$ array with the $K$ and $Q$ value for the accelerometers increased by a factor of 10 and decreased by a factor of 10 to test the effect of sensor quality. Each run is conducted with the exact same random noise seed to better observe the effect of the noise power.
Table 4.3: 3σ Error for Various Noise Coefficients

<table>
<thead>
<tr>
<th>Coefficients</th>
<th>3σ Error Angle (deg)</th>
<th>3σ Rate Error (deg/s)</th>
<th>Max Angle Error (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>×0.1</td>
<td>1.12</td>
<td>1.09</td>
<td>1.94</td>
</tr>
<tr>
<td>default</td>
<td>1.22</td>
<td>1.47</td>
<td>1.70</td>
</tr>
<tr>
<td>×10</td>
<td>2.87</td>
<td>23.2</td>
<td>4.54</td>
</tr>
</tbody>
</table>

As the noise power increased, so did the error in all three measurements in Table 4.3. This means that the Virtual Gyroscopes noise did have some effect on the filter’s performance; it was not completely ignored. If more noise is present in the accelerometers, more noise will be present in the angular velocity measurement. Therefore, when the angular velocity was updated by the estimated bias, a large noise power in the measurement led to a larger noise power in the estimate. Because the same noise seed was used for each test and the array size was constant, it was not necessary to perform longer simulation runs to see the effect of the changing noise coefficients. As noise increased, the rate error in the angular velocity estimate was the most affected. The angle 3σ and max angle error also increased, although by a smaller factor.

4.3.3 Temperature Fluctuation

The test with $R_{14}$ was repeated, except this time temperature was allowed to vary.
Figure 4.6: Filter Performance with Variable Temperature

From Figure 4.6, there was not a significant change in filter performance as the temperature varied. This again was an indicator that the Virtual Gyroscope measurement was not actually improving filter performance.

4.3.4 Reexamining Filter Tuning

After finding that neither array size nor temperature had a significant effect on the filter’s performance with the current tuning of the filter, the methodology required
a closer look. It seemed that more of the improvement in measurement quality was due to the filtered attitude measurement than the filtered rate measurement. Was the Virtual Gyroscope actually helping, or was performance improvement due to the filtering of the measurements from the attitude sensor alone? To test this, the output of the Virtual Gyroscope was intentionally set to an incorrect constant, and the simulation was repeated.

The Virtual Gyroscope output was set to the vector $\tilde{\omega} = \begin{bmatrix} 1 & -1 & 0 \end{bmatrix}^T$. Figure 4.7 shows the comparison of the angle error and rate error with and without the Virtual Gyroscope in use.
It was found that the performance of the attitude filter did not decrease when the Virtual Gyroscope measurement was ignored entirely. The current values $\sigma_R$ in relation to $\sigma_u$ and $\sigma_v$ meant that the filter simply overwrote the measured angular velocity at each bias update step.

This effect was demonstrated by setting the gyroscope estimate to near zero and plotting the estimate of the bias $\hat{\beta}$ alongside the true angular velocity. Even though the measurement from the gyroscope indicated the attitude quaternion should be a
constant, the filter forced the estimated angular velocity to agree with the measured quaternion.

![Filter Performance with Rate Estimate Set to Zero](image)

**Figure 4.8: Estimating Angular Velocity with the Bias Update**

In Figure 4.8, the negative of the estimated bias is plotted in red with the true angular velocity blue. The solid black line shows the incorrect value that the Virtual Gyroscope output was set to for this test case.

### 4.3.5 Forcing the Filter to Use the Virtual Gyroscope

This section examines the effect on the filter’s performance if the filter is tuned to rely more heavily on the Virtual Gyroscope estimate. This can be accomplished simply by increasing $\sigma_R$ to decrease the filter’s “trust” of the attitude measurement.
\( \sigma_R \) is increased by a factor of 100, and the bias is plotted again with the VG output set near zero.

![Filter Performance with Rate Estimate Set to Zero](image)

**Figure 4.9: Effect of Increased \( \sigma_R \)**

As can be seen from the red line on Figure 4.9, the bias no longer forces the angular velocity estimate to the value predicted based on the attitude measurement, but instead lags behind the true angular velocity due to the incorrect information from the gyroscope. The test case from Table 4.1 is repeated with the Virtual Gyroscope active once again, but this time with the increased \( \sigma_R \).

Previously, the 3\( \sigma \) error in the attitude estimate was 1.07°. With an increased reliance on the Virtual Gyroscope, the error in the attitude estimate increased to 52°. If \( \sigma_R \) is only increased by a factor of 10, the angle error doesn’t increase quite as dramatically, but still rises 3°.
The same effect can be accomplished by leaving $\sigma_R$ at the numerically estimated value in Table 4.1 but decreasing $\sigma_u$ and $\sigma_v$ instead. In this case the resulting angle error is 3.6°.

Based on these tests, it was seen that the attitude filter can be tuned to rely more heavily on the Virtual Gyroscope, but doing so strictly decreased the estimate’s quality. The best performance for the attitude filter with attitude measurements available at the same pace as angular velocity measurements remains the configuration in Table 4.1.

4.4 Testing the Filter with Less Frequent Attitude Measurements

It may be unrealistic for many CubeSat missions to assume that attitude measurements with 5 degree error are available to the attitude filter at 20 Hz. This section examined the effect on filter performance using the nominal array configuration again, but this time the attitude measurement frequency was increased to 1 Hz. The value of $\sigma_R$ is set to a 100 times the nominal value to make sure the measured angular velocity has more influence on the filtered result. This decrease makes the $R$ matrix no longer reflect the measured value, but instead artificially places more confidence in the gyro measurement, while the true error covariance due to the noise added to the quaternion remains unchanged. Figure 4.10 shows the effect of low frequency attitude measurements on the quaternion estimate.
Figure 4.10: Less Frequent Attitude Measurements with More Trust in VG

Because the attitude estimate is only used once per second, the estimate from the Virtual Gyroscope must be used to propagate the quaternion between measurements. Every one second, a new attitude estimate arrives and pushes the estimate of the quaternion closer to the true value. As is evident from the graph, trusting the Virtual Gyroscope more is a bad plan for improving filter performance.

If the value of $\sigma_R$ is left at the nominal value, which indicate that the attitude measurement is more trustworthy than the VG, the estimate improves significantly.
As shown in Figure 4.11, even with attitude estimates with five degrees of error available only once a second, it is still better to place little confidence in Virtual Gyroscope estimate by keeping $\sigma_R$ small.

4.4.1 Comparison to a MEMS Gyroscope

The MEKF successfully filters the attitude when the contribution of the Virtual Gyroscope is kept small, even at a low attitude sensor sample rate. This is, in part, due to the fact that the control torque applied for the prior test cases had a low period and magnitude, so the changes in the angular velocity were small. In this section, the period of the sinusoidal control torque is decreased from 1000 to 50 seconds. The performance of the attitude filter will be compared using the Virtual Gyroscope and a MEMS gyroscope model. The MEMS gyroscope will be simulated very simply by adding white noise and bias with a rate of change of white noise, using the nominal
standard deviation values from Table 4.1. This corresponds to a gyro with similar noise power to the Virtual Gyroscope, but bias dynamics that are more similar to the expected gyro bias for the filter.

The measurement equations for the MEMS gyro become:

\[
\bar{\omega} = \omega + \eta_u + \beta, \quad (4.34)
\]

where \(\eta_u\) is Gaussian white noise with variance \(\sigma_u^2\) and \(\dot{\beta} = \eta_v\) where the variance of \(\eta_v\) is \(\sigma_v^2\). Six tests runs are conducted in total, three for each sensor, with the values of \(\sigma_R\) tuned to trust the gyro less, nominally, and more (.01\(\sigma_R\), \(\sigma_R\), and 100\(\sigma_R\)) respectively.

The resulting angle errors are shown in Table 4.4.

<table>
<thead>
<tr>
<th>Gyro Confidence</th>
<th>Angle Error with VG (deg)</th>
<th>Angle Error with MEMS (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>7.48</td>
<td>7.3</td>
</tr>
<tr>
<td>Nominal</td>
<td>10.5</td>
<td>4.2</td>
</tr>
<tr>
<td>High</td>
<td>197</td>
<td>28</td>
</tr>
</tbody>
</table>

As can be seen from Table 4.4, the MEMS gyroscope with the same effective noise gyro coefficients as the Virtual Gyroscope performs better for every tested confidence, most notably when the filter uses the measured value of \(\sigma_R\) instead of artificially increasing the confidence in the attitude sensor. This indicates that for the Virtual Gyroscope, it is better to trust the attitude sensor even at low sample rates. For the MEMS gyroscope, it is better to tune to filter to the measured \(\sigma_R\), instead of artificially placing more confidence in the attitude sensor.
4.5 Chapter 4 Conclusions

The main conclusion from this chapter is that the Multiplicative Extended Kalman Filter is a powerful tool for attitude filtering, having improved the performance of the attitude sensor alone by a factor of five, even when run at a low frequency. For many attitude and rate estimation purposes, an extended Kalman filter and an attitude sensor without an angular velocity measurement device is a quite viable choice for CubeSat missions.

The Virtual Gyroscope, however, did not improve the attitude estimate at all. This is a strong case against the use of an accelerometer array in place of a typical gyroscope for CubeSat attitude determination. If it is only necessary to determine attitudes and angular rates within a few degrees or degrees per second of uncertainty, attitude sensors and an extended Kalman filter provide ample quality estimates. If a higher fidelity estimate is required, either a better attitude sensor like a star tracker will be necessary, or a high quality standard gyroscope. Whether or not the Virtual Gyroscope has use cases for nanosatellite attitude determination, whether with different sensor quality or a more sophisticated algorithm, remains an open question. It is possible a different filter architecture, perhaps one that resolves the bias of each accelerometer before inputting it into the accelerometer array, could have better performance.
Chapter 5

CONCLUSION

The main objective of this thesis was to create and qualify the “Virtual Gyroscope” to replace a standard MEMS gyroscope for attitude estimation on nanosatellites. The mathematical derivation of the Virtual Gyroscope algorithm was presented, building off the results from Hamley and Crassidis in [11], along with an analysis of the error characteristics of the angular velocity measurement due to the error of the accelerometers in the array. To test the Virtual Gyroscope, a simple approximation of the physics of a 1U CubeSat was simulated, along with a MEMS accelerometer model that was capable of reproducing the stochastic error characteristics of a high quality MEMS sensor. This model was validated using Root Allan Variance, and used to provide the input to the Virtual Gyroscope algorithm. After studying the output of the VG, both using Root Allan Variance and more qualitative approaches, the Virtual Gyroscope was paired with a Multiplicative Extended Kalman Filter to evaluate its performance for CubeSat attitude estimation.

The key limitation of the Virtual Gyroscope algorithm is that it cannot predict and eliminate the bias of the individual accelerometer elements that make up the array. It was demonstrated in chapters 2 and 3 that the bias of the Virtual Gyroscope is a combination of the unknown bias of terms of each sensor in the array. This weakness was exacerbated by the fact that the distance from the axis of rotation and the magnitude of the angular velocity experienced by CubeSats are relatively small, so the centripetal acceleration due to the rotation was difficult to detect through the noise.
When paired with the Multiplicative Extended Kalman Filter, it was found that the angular velocity estimate output by the Virtual Gyroscope were too poor to improve the filter’s performance, even when the attitude sensor delivered low frequency measurements with added noise. Even with a low sampling rate and high measurement noise, the Kalman filter and attitude sensor outperformed the Virtual Gyroscope as a tool for estimating angular velocity and attitude, regardless of the size and noise power of the accelerometer array.

5.1 Potential Improvements

It is clear from this research that as long as the bias of the accelerometers is unknown, the measurement from the Virtual Gyroscope will be too poor to improve the filter’s performance. It is possible to calculate the expected measurement of each accelerometer based on an estimate of the angular velocity and acceleration of the spacecraft. This means that an alternate Kalman filter architecture that predicts the bias of each individual accelerometer instead of attempting to predict the bias of the gyroscope as a whole could yield significant improvements in performance. Such an approach would still require an attitude sensor, but if the angular velocity estimates are low-noise, then the quality of the sensor needed to meet the performance requirements would be reduced.

For many applications, a well-tuned Kalman filter along with an attitude measurement could also be an effective Virtual Gyroscope, depending on the quality and sampling rate of the attitude sensor. Because of the poor results of the current Virtual Gyroscope algorithm, it was difficult to evaluate other performance metrics. An alternate Virtual Gyroscope and sensor fusion algorithm may be able to use an accelerometer array more effectively to resist random walk and temperature drift,
improve resiliency to sensor failure, and add a new option for CubeSat attitude estimation schemes.
BIBLIOGRAPHY


This appendix includes relevant mathematical facts and derivations that were not included elsewhere in the text.

This description of Root Allan Variance (RAV) testing was adapted from [10]. Allan Variance (AV) is one of the simplest techniques for stochastic modeling of inertial sensor errors. This method calculates the root means square (RMS) random drift error for different averaging times of the output signal. The RAV curves can be used to determine the characteristics of the random processes that can be used to simulate measurement noise.

A.1 Power Spectral Density (PSD)

Power spectral density is a method of representing the covariance of a stationary random process in the frequency domain, where the relationship between them is given by

\[ S(\Omega) = \int_{-\infty}^{\infty} e^{-j\Omega \tau} K(\tau) d\tau, \]

where \( S(\Omega) \) is the Fourier transform of the covariance as a function of time. Here \( \Omega \) is reserved for angular frequency (because \( \omega \) is already taken by angular velocity), \( K(\tau) \) is the covariance of the signal, and \( j \equiv \sqrt{-1} \).
For linear systems, a transformation applied to a random process can be represented using the transfer functions in the frequency domain. The PSD of the output process is given by

\[ S_{\text{out}}(\Omega) = H(j\Omega)S_{\text{in}}(\Omega)H^*(j\Omega), \]  

(A.2)

where \( H \) is the transfer function, \( S_{\text{out}} \) and \( S_{\text{in}} \) are the output and input PSD respectively, and \( H^* \) is the complex conjugate transpose of \( H \).

### A.2 Calculating Allan Variance

If the input is assumed to be white noise (\( S_{\text{in}} = \text{constant} \)), then the output PSD is defined by the transfer function. It is a common assumption that the noise processes in a signal are the result of some shaping filter transfer function acting on white noise, where the shaping filter is chosen in such a way that the statistics of the output spectrum match the observed statistics for the actual device. Using this idea, Allan Variance provides a method for determining the coefficients of the transfer functions, which is necessary for this thesis to accurately simulate sensor noise.

The three noise terms that are of interest for accelerometer and Virtual Gyroscope simulation are (for a gyroscope) angle random walk, bias instability, and rate random walk. For an accelerometer, the terms are called velocity random walk, bias instability, and acceleration random walk. The method for determining the coefficients of each process is agnostic to the type of sensor, so only the process for a gyroscope is shown here.

To calculate Allan Variance, \( N \) consecutive data points are collected at a sample time \( t_0 \). These data points are then partitioned into many smaller clusters of size \( n < N/2 \).
The length of each cluster is $T = nt_0$. The cluster average is defined as

$$\bar{\omega}_k(T) = \frac{1}{T} \int_{t_k}^{t_k+T} \omega(t) dt$$

(A.3)

where $\omega(t)$ is the output angular velocity of a gyroscope. Equation (A.3) is the average angular velocity for the cluster beginning at the $k$th data point.

The cluster of data of the same length that begins $n$ data points later also has a cluster average defined as

$$\bar{\omega}_{next}(T) = \frac{1}{T} \int_{t_{k+1}}^{t_{k+1}+T} \omega(t) dt$$

(A.4)

where $t_{k+1} = t_k + T$.

The difference between the two cluster averages is

$$\xi_{k+1,k} = \bar{\omega}_{next}(T) - \bar{\omega}_k(T).$$

(A.5)

For every possible cluster time $T$, an ensemble of $\xi$ values forms a set of random variables, where the variance of that set is the Allan Variance.

The Allan Variance for an integration time of $T$ is defined as

$$\sigma^2(T) = \frac{1}{2(N - 2n)} \sum_{k=1}^{N-2n} (\bar{\omega}_{next}(T) - \bar{\omega}_k(T))^2.$$  

(A.6)

Because $N$ is finite, equation (A.6) is only an estimate of the true variance. The more independent clusters that are formed, the more accurate the estimate.
By integrating the angular velocity measurement from the sensor after some initial time, Allan Variance can be defined using angles.

\[ \theta(t) = \int_0^t \omega(t) \, dt. \]  

(A.7)

The time \( t \) will be integer multiples of the sampling time \( t = kt_0 \), for \( k = 1, 2, 3, \ldots, N \).

For an accelerometer, velocity is used instead of angle. The zero point for the lower bound of the integral can be chosen arbitrarily because only the difference in \( \theta \) is used in the following equations. Letting \( \theta_k = \theta(kt_0) \), equation (A.6) is rewritten in terms of the angle by substitution.

\[ \bar{\omega}_k(T) = \frac{\theta_{k+n} - \theta_k}{T}, \]  

(A.8)

and

\[ \bar{\theta}_{\text{next}}(T) = \frac{\theta_{k+2n} - \theta_{k+n}}{T}, \]  

(A.9)

so

\[ \sigma^2(T) = \frac{1}{2T^2(N-2n)} \sum_{k=1}^{N-2n} (\theta_{k+2n} - 2\theta_{k+n} + \theta_k)^2. \]  

(A.10)

The power spectral density of the random process with variance \( \sigma^2(T) \) can be related to the power spectral density of the random process of angular velocities at each integration time \( \omega(T) \). That is,

\[ \sigma^2(T) = 4 \int_0^\infty S_\omega(f) \frac{\sin^4(\pi f T)}{\pi f T} \, df. \]  

(A.11)

If a power spectral density function is substituted into equation (A.11), an expression for Allan Variance as a function of the averaging time \( T \) can be determined. The numerically derived Allan Variance curve from experimental data is compared to the
analytical expression for $\sigma^2(T)$ with a chosen $S_\omega$. From this comparison, the influence of that particular spectral density function on the sensor’s signal can be evaluated.

Consider, for example, white noise, which has a constant power spectral density at all frequencies. The transfer function for this contribution is the white noise coefficient squared, which is also referred to as the angle random walk (for a gyroscope) or the velocity random walk (for an accelerometer).

$$S_\omega = Q^2$$  \hspace{1cm} (A.12)

where $Q$ is called the white noise coefficient. If this transfer function is substituted into equation (A.11), the expected Allan Variance has the analytical expression

$$\sigma^2(T) = \frac{Q^2}{T}.$$  \hspace{1cm} (A.13)

On a log-log plot of $\sigma(T)$ versus $T$, this will be a line with a slope of $-1/2$. So to estimate the value of $Q$ from numerical data, one must examine the square root of Allan Variance versus integration time at $T = 1$.

Similar substitutions are made for bias instability, which has PSD

$$S_\omega(f) = \begin{cases} \left(\frac{B^2}{2\pi}\right) \frac{1}{f}, & f \leq f_0, \\ 0, & f > f_0. \end{cases}$$ \hspace{1cm} (A.14)

The analytical expression for Allan Variance with this PSD is approximately

$$\sigma^2(T) = \frac{2B^2}{\pi},$$ \hspace{1cm} (A.15)

which is a constant with respect to integration time. Therefore, the coefficient $B$ can be taken from the RAV curve where the slope is zero. Finally, rate random walk (for
a gyroscope) or acceleration random walk (for an accelerometer) has the PSD

\[ S_\omega(f) = \left(\frac{K}{2\pi}\right)^2 \frac{1}{f^2}, \]  
(A.16)

where K is called the rate random walk coefficient or the acceleration random walk coefficient. Substituting this power spectral density in (A.11) yields

\[ \sigma^2(T) = \frac{K^2 T}{3}, \]  
(A.17)

so K can be estimated with the +1/2 on the log-log plot.

For real sensors there will be multiple additive noise contributions whose effects are larger for different integration times. As such, none of these three slopes (-1/2, 0, or 1/2) will be present for the whole curve like in Figure 3.2. There are two options to estimate the coefficients from an RAV curve in this case. The first option (and the one employed in this thesis), is to fit a line with the appropriate slope to the RAV curve, then evaluate the linear fit at an easy value of T. For white noise, the obvious value to choose is \( T = 1 \) because \( \sigma(1) = Q \) based on equation (A.13). For rate/acceleration random walk, choose \( T = 3 \), in which case \( \sigma^2(3) = K \) from equation (A.17). Alternatively, any point can be used along the -1/2 or +1/2 portion of the RAV, and the measured values \( \sigma \) and the correspond \( T \) can be plugged into equations (A.13) and equation (A.17). The resulting expression can then be solved for Q and K.

To evaluate Allan Variance from a set of data, the allanvar function from Mathworks was used [3].