LONG TERM QUADROTOR STABILIZATION

THESIS

Nicolas Hamilton, Second Lieutenant, USAF

AFIT/GE/ENG/11-13

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT/GE/ENG/11-13

LONG TERM QUADROTOR STABILIZATION

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Nicolas Hamilton, BSEE

Second Lieutenant, USAF

March 2011

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT/GE/ENG/11-13

LONG TERM QUADROTOR STABILIZATION

Nicolas Hamilton, BSEE
Second Lieutenant, USAF

Approved:

Lt Col Michael Stepaniak (Chairman)	Date
Dr. Meir Pachter (Member)	Date
Maj Kenneth Fisher (Member)	Date

AFIT/GE/ENG/11-13

Abstract

One major research thrust of the Advanced Navigation and Technology (ANT) Center at the Air Force Institute of Technology (AFIT) is navigating in environments where signals from global navigation satellite systems (GNSS), such as the Global Positioning System (GPS), are either unavailable or unusable due to low signal strength or multipath effects. These environments include indoor, urban, and underground environments and can include environments where these signals are being actively denied either intentionally or unintentionally. The ANT Center has identified small, hovering, unmanned air vehicles (UAVs) as one of the best solutions since they offer greater mobility in these environments than fixed wing UAVs or unmanned ground vehicles. Quadrotors were selected over more traditional helicopter configurations because of their mechanical simplicity. In order to navigate in these type of environments, an inertial measurement unit (IMU) is used and typically augmented with other sensors.

The work of this thesis focuses on achieving better IMU long term stability and a better navigation solution. This is done in two ways. First, the IMU accelerometer output is examined to determine if it is possible to use accelerometers to determine attitude. If the quadrotor is stationary or moving at constant velocity, the roll and pitch angles can be determined. Additionally, the accelerometers can be used to determine angular accelerations and angular rates which are integrated to determine heading, but this is impractical with current technology. The second approach models the quadrotor and uses the models in Kalman Filters along with the IMU measurements to determine the best possible navigation solution.

Acknowledgements

First and foremost I thank God for everything I have and all the opportunities I have been given. I also appreciate my parents, sisters, and brother for their support throughout my life and while here at AFIT. I'm also thankful for the Air Force for providing me the oppurtunity to get both my bachelor and master's degrees (or at least I hope I get the master's degree), at little expense to me.

I appreciate the work of my advisor, Michael Stepaniak, which was instrumental in getting me started on this project. I also appreciate his patience in helping me to understand a number of concepts that I was unfamiliar with, especially early on. I really liked it when he turned me on to a wrong path with accelerometers (they measure g-forces non-techies) early on and we went back and forth over which one of us was right for several months. I really enjoyed those conversations in his office, even though I was sweating bullets since I was trying to tell a superior officer he was wrong. I'm also thankful for all the help and support provided by the interns and staff of the Advanced Navigation and Technology Center; Chris Lomanno, Jared Kresge, Tom Pestak, Mark Smearcheck, and Fred Webber (in alphabetical order, not order of helpfulness, I'll let them fight over who was most helpful). Many of the students also enrolled in the guidance, navigation, and control course of study within AFIT's electrical engineering department were a big help to me and each other as we muddled through the classes and our respective theses by bouncing thoughts and ideas around. I'd also like to thank the academy... just kidding. If you're not a tech geek like me, you can read the abstract and intro and skip the rest.

Nicolas Hamilton

Table of Contents

Page

Abstract iv

Acknowledgements v

List of Figures viii

List of Tables xii

List of Symbols xiii

List of Acronyms xix

I. Introduction 1

1.1 Problem Description 1
1.2 Proposed Solution 2
1.3 Chapter Summary 2

II. Background 4

2.1 Quadrotor History 4
2.2 Quadrotor Operation 5
2.3 Equations Governing Motion and Measurements 7
2.3.1 Equations of Motion 7
2.3.2 Measurement Equations 8
2.3.3 Forces and Moments 12
2.4 Control Scheme 15
2.5 Controller Implementation 18

III. Quadrotor Development 19

3.1 Structural Configuration 19
3.2 Electronics Configuration 26

IV. Methodology 30

4.1 Development 30
4.2 Linear Model 34
4.2.1 Development 34
4.2.2 Validation 36
4.3 Measurement Model Development 36
4.3.1 Noise Characterization 36
4.3.2 Kalman Filter 37

Page

V. Results and Analysis 41

5.1 Measurement Noise Characterization Results 41
5.2 Non-linear Model Simulation Results 49
5.3 Linear Model Validation Against Non-Linear Model 56
5.4 Kalman Filter Performance 68
5.5 Using Accelerometers to Determine the Gravity Vector 72
5.6 Using Accelerometers to Measure Angular Rates Results 75
5.7 Periodic Landing to Improve Attitude Solution 86

VI. Conclusion 93

Bibliography 96

List of Figures

Figure		Page
1	Quadrotor Viewed From Above With Z-Axis Directed Down Into The Page.	5
2	Quadrotor Rolling.	6
3	Quadrotor Pitching.	6
4	Quadrotor Yawing.	6
5	2-Dimensional Quadrotor (Bi-rotor) With X and Y axis Accelerometers at Each End.	11
6	Controller Verification.	16
7	Roll controller surrounding the quadrotor roll channel plant model.	16
8	Pitch controller surrounding the quadrotor pitch channel plant model.	17
9	Yaw controller surrounding the quadrotor yaw channel plant model.	17
10	Quadrotor Represented As A Point Cloud To Compute Center Of Gravity And Moments Of Inertia.	25
11	Previous Quadrotor Represented As A Point Cloud.	26
12	Electronics Configuration.	28
13	48 Hour IMU Data Collect, Gyro Allan Variance Analysis Results.	42
14	48 Hour IMU Data Collect, Accelerometer Allan Variance Analysis Results.	43
15	Gyro Allan Variance Bias Instability Estimation.	44
16	Accelerometer Allan Variance Bias Instability Estimation.	45

Figure Page

17 Power Spectral Density of Collected and Simulated Data for x-axis Gyro. 46

18 Power Spectral Density of Collected and Simulated Data for y-axis Gyro. 46

19 Power Spectral Density of Collected and Simulated Data for z-axis Gyro. 47

20 Power Spectral Density of Collected and Simulated Data for x-axis Accelerometer. 47

21 Power Spectral Density of Collected and Simulated Data for y-axis Accelerometer. 48

22 Power Spectral Density of Collected and Simulated Data for z-axis Accelerometer. 48

23 Gyro Allan Variance Bias Instability Estimation and Simulated Data. 49

24 Accelerometer Allan Variance Bias Instability Estimation and Simulated Data. 50

25 Simple Non-Linear Model Static Simulation Results with Standard Mixer Matrix. 52

26 Simple Non-Linear Model Dynamic Simulation Results with Standard Mixer Matrix. 53

27 Complex Non-Linear Model Static Simulation Results with Standard Mixer Matrix. 54

28 Complex Non-Linear Model Dynamic Simulation Results with Standard Mixer Matrix. 55

29 Linear Model Static Simulation Results with Standard Mixer Matrix. 57

30 Linear Model Dynamic Simulation Results with Standard Mixer Matrix. 58

Figure Page

31 Linear Model Static Simulation Results with Standard Mixer Matrix and No Noise. 59

32 Linear Model Dynamic Simulation Results with Standard Mixer Matrix and No Noise. 60

33 Linear Model 2 Static Simulation Results with Standard Mixer Matrix. 61

34 Linear Model 2 Dynamic Simulation Results with Standard Mixer Matrix. 62

35 Linear Model 3 Static Simulation Results with First Modified Mixer Matrix. 65

36 Linear Model 3 Dynamic Simulation Results with First Modified Mixer Matrix. 66

37 Linear Model 4 Static Simulation Results with Second Modified Mixer Matrix. 68

38 Linear Model 4 Dynamic Simulation Results with Second Modified Mixer Matrix. 69

39 Linear Model 1 Static Simulation Results with Standard Mixer Matrix and Kalman Filter On. 70

40 Linear Model 1 Dynamic Simulation Results with Standard Mixer Matrix and Kalman Filter On. 71

41 Linear Model 4 Static Simulation Results with Second Modified Mixer Matrix and Kalman Filter On. 72

42 Linear Model 4 Dynamic Simulation Results with Second Modified Mixer Matrix and Kalman Filter On. 73

43 Using Accelerometers to Determine the Roll and Pitch Angles while Stationary. 75

44 Using Accelerometers to Determine the Roll and Pitch Angles while Moving. 76

45 Centripetal Acceleration Method of Determining Angular rate vs Using Gyros with a simulated 1 degree per second angular rate and 5mg accelerometer noise. 79

Figure Page

46 Centripetal Acceleration Method of Determining Angular rate vs Using Gyros with a simulated 1 degree per second angular rate and 5mg accelerometer noise. 80

47 Centripetal Acceleration Method of Determining Angular rate vs Using Gyros with a simulated 300 degree per second angular rate and 5mg accelerometer noise. 82

48 Centripetal Acceleration Method of Determining Angular rate vs Using Gyros with a simulated 300 degree per second angular rate and 5mg accelerometer noise. 83

49 Centripetal Acceleration Method of Determining Angular rate vs Using Gyros with a simulated 100 degree per second angular rate and 500ug accelerometer noise. 84

50 Centripetal Acceleration Method of Determining Angular rate vs Using Gyros with a simulated 100 degree per second angular rate and 500ug accelerometer noise. 85

51 Centripetal Acceleration Method of Determining Angular rate vs Using Gyros with a simulated 1 degree per second angular rate and 5ng accelerometer noise. 87

52 Angular Acceleration Method of Determining Angular rate vs Using Gyros with a simulated 1 degree per second angular rate and 5ng accelerometer noise. 88

53 Periodically Landing Quadrotor on Flat Plane. 89

54 Quadrotor Attempting to Hover. 90

55 Periodically Landing Quadrotor on an Inclined Plane. 91

List of Tables

Table		Page
1	Component Mass, Size, and Position Properties.	21
2	Component Mass, Size, and Position Properties Continued.	22
3	Accelerometer Noise Standard Deviation vs. Angular Rate Threshold and Accelerometer Technology	87

List of Symbols

Symbol	Definition
$\vec{a}_{meas}$	Accelerometer measurement vector
a_x	x-axis accelerometer measurement
a_y	y-axis accelerometer measurement
a_z	z-axis accelerometer measurement
A	State matrix
B	Input matrix
B_d	Discrete input matrix
$counts$	Field progammable gate array commands
C_b^n	Direction cosine matrix from body frame to navigation frame
C_n^b	Direction cosine matrix from navigation frame to body frame
C_Q	Rotor torque coefficient
C_T	Rotor thrust coefficient
CG_x	x-axis center of gravity offset
CG_y	y-axis center of gravity offset
CG_z	z-axis center of gravity offset
d	Distance of rotors from center of vehicle
E	Expectation operator
$\vec{F}$	Total force vector
F_i	Thrust generated by the i^{th} rotor
F_{thrust}	Total quadrotor thrust
$\vec{F}_{thrust}$	Quadrotor thrust vector
F_T	Rotor thrust force
F_{Tref}	Reference thrust force

Symbol	Definition
F_x	x-axis force
F_y	y-axis force
F_z	z-axis force
g	Gravitational acceleration
H	Measurement matrix
H_{arx}	x-axis gyro transfer function
H_{ary}	y-axis gyro transfer function
H_{arz}	z-axis gyro transfer function
H_{ax}	x-axis accelerometer transfer function
H_{ay}	y-axis accelerometer transfer function
H_{az}	z-axis accelerometer transfer function
H_{lead}	Lead compensator transfer funtion
H_{motor}	Motor transfer function
$H_{sampler}$	Sampler transfer function
i	An integer indexing variable
I_x	x-axis moment of inertia
I_y	y-axis moment of inertia
I_z	z-axis moment of inertia
I_{xy}	xy-axis moment of inertia
I_{xz}	xz-axis moment of inertia
I_{yz}	yz-axis moment of inertia
j	An integer indexing variable
k	Discrete time index
L	x-axis moments

Symbol	Definition
m	Quadrotor mass
m_i	Mass of a point mass
M	y-axis moments
Mix	Mixer matrix
n	Total number of point masses m_i
n_x	x-axis normalization factor
n_y	y-axis normalization factor
N	z-axis moments
p	Body frame x-axis angular rate
$\dot{p}$	Body frame x-axis angular acceleration
P	State estimate covariance
PWM	Pulse width modulation Commands
q	Body frame y-axis angular rate
$\dot{q}$	Body frame y-axis angular acceleration
Q	Process noise covariance
r	Body frame z-axis angular rate
$\dot{r}$	Body frame z-axis angular acceleration
R	Measurement noise covariance
R_l	Rotor blade radius (ft)
s	Laplace transfrom variable
Δt	Simulation time step
Δt_m	Measurement time step
u	x-axis velocity
$\dot{u}$	x-axis acceleration

Symbol	Definition
$\vec{u}$	Input vector
u_i	i^{th} rotor command
u_p	Roll command
u_q	Pitch command
u_r	Yaw command
u_{thrust}	Thrust command
v	y-axis velocity
$\dot{v}$	y-axis acceleration
$\vec{v}$	Measurement noise vector
w	z-axis velocity
$\dot{w}$	z-axis acceleration
$\vec{w}$	Process noise vector
x	x position
$\vec{x}$	State vector
$\dot{\vec{x}}$	Derivative of the state vector
x_i^b	x position of i^{th} point mass m_i in the body frame
$x_b(i)$	x position of the i^{th} rotor in the body frame
y	y position
y_i^b	y position of i^{th} point mass m_i in the body frame
$y_b(i)$	y position of the i^{th} rotor in the body frame
z	z position
z_i^b	z position of i^{th} point mass m_i in the body frame
$\vec{z}$	Measurement vector

Symbol	Definition
Greek Symbols	
θ	y-axis Euler Angle (Pitch)
$\dot{\theta}$	y-axis Euler Angle rate
ρ	Air density
τ	Torque
τ_i	Torque generated by the i^{th} rotor ($lbf \cdot ft$)
ϕ	x-axis Euler Angle (Roll)
$\dot{\phi}$	x-axis Euler Angle rate
Φ	State transition matrix
ψ	z-axis Euler Angle (Yaw)
$\dot{\psi}$	z-axis Euler Angle rate
ω	Rotor angular speed
Subscripts and Superscripts	
b	Body frame
i	Integer index
j	Integer index
int	Intermediate values calculated within control loops
n	Navigation frame
$true$	Truth value
x	Directed along the x-axis
y	Directed along the y-axis
z	Directed along the z-axis
$-$	Before measurement update

$+$ After measurement update

List of Acronyms

Acronym	Definition
AFIT	Air Force Institute of Technology
ANT	Advanced Navigation Techonology
BEC	Battery Eliminator Circuit
DCM	Direction Cosine Matrix
ESC	Electronic Speed Controller
FPGA	Field Programmable Gate Array
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
IMU	Inertial Measurement Unit
LADAR	Laser Detection and Ranging
LiPo	Lithium-Polymer
LQR	Linear Quadratic Requlator
MEMS	Micro-Electro-Mechanical Systems
PWM	Pulse Width Modulation
PSD	Power Spectral Density
UAV	Unmanned Air Vehicle

LONG TERM QUADROTOR STABILIZATION

I. Introduction

1.1 Problem Description

There is a desire in the Air Force, and at the Advanced Navigation and Technology (ANT) Center in the Air Force Institute of Technology (AFIT) in particular, to navigate in indoor, urban, and underground environments. Additionally, it is desired that there should be unmanned vehicles to fill this role, particularly vehicles that have a great degree of mobility. As a result, hovering unmanned air vehicles (UAVs) have been selected as the vehicles of choice in these environments. It is also desired that these vehicles be able to navigate in the absence of any type of global navigation satellite system (GNSS) because these environments make obtaining usable GNSS data difficult. This goal would also allow navigation in areas where GNSS service is denied. The primary navigation tool used in these type of environments is an inertial measurement unit (IMU). While IMUs provide very accurate short term navigation, they are very poor at long term navigation, especially the smaller micro-electro-mechanical systems (MEMS) grade IMUs that must be used on the small UAVs designed to go into these environments. As a result, it is desirable to augment an IMU with additional sensors to navigate. Unfortunately, adding more sensors results in a larger payload for the UAVs. Most UAVs that could possibly fill this desired role either rely on the Global Positioning System (GPS), have a small payload capacity, or both [1, 2, 12, 14, 25, 22]. This thesis will primarily extend the work of Michael Stepaniak which was focused on this same problem and resulted in the construction

and flight of a quadrotor with a 10 pound payload capacity. It is the goal of this thesis to develop ways to navigate with the IMU alone for longer time periods so that it will be a more stable platform to integrate with other sensors in the future.

1.2 Proposed Solution

There are two primary ways in which this problem will be approached. The first is to start with the IMU and examine the data it outputs to determine if there is a way to use other information from the IMU to achieve a better navigation solution. This approach will look at using the accelerometers to determine the quadrotor's roll, pitch, and yaw as opposed to using the gyros. The second is to examine ways to better model the quadrotor. If the quadrotor can be modeled better, the IMU measurements can be integrated with model based predictions of how the quadrotor should behave to determine a better navigation solution than either could offer alone. In particular, non-linear models will be developed to best capture the quadrotor's dynamics for simulation purposes, then linear models will be developed based on these non-linear models. The linear models can then be integrated with IMU measurements using a Kalman Filter in an attempt to find a better navigation solution than the IMU measurements alone.

1.3 Chapter Summary

Chapter 2 begins with a brief look at the history of quadrotors. Then the general operation of quadrotors and the specific configuration of the quadrotor used in this thesis are covered. Finally, the equations of motion relevant to the quadrotor are examined and along with the control scheme being used on the quadrotor. Chapter 3 describes how the non-linear and linear models are developed and validated. It also presents modeling the IMU measurement noise and how the quadrotor is tested.

Chapter 4 then examines the results of those models and tests through figures and analysis. Chapter 5 summarizes the results, offers conclusions, and suggests areas for future work.

II. Background

2.1 Quadrotor History

The first quadrotor was developed by Louis and Jacques Bréguet with help from their professor Charles Richet and its first flight was either on August 24th or September 7th 1907, the exact date being ambiguous. It was dubbed the Gyroplane No. 1 and had the distinction of being the first manned helicopter; however, it never officially flew because it was tethered and also stabilized by ground crew [19].

Quadrotors were largely ignored until recently when researchers began looking at them as potential platforms for small and micro unmanned air vehicles (UAVs) and also as an interesting controls problem. Much of the research has been directed at trying various control schemes to augment an inertial measurement unit (IMU) with other sensors. Some of the control schemes attempted include classical methods such as tuning proportional, integral, and derivative control schemes [1, 2, 12, 14, 25, 22, 23] along with combinations of these and also pole placement through use of Root-Locus [1]. Some other control schemes that have been applied include model based predictive control (MBPC) [5], H-infinity [1], neural networks [22], linear quadratic regulator (LQR) [1, 5], and a variety of non-linear approaches [14].

Sensors that have been used to augment an IMU include cameras [1, 9, 12, 14], laser range finders that use laser detection and ranging (LADAR) and scanning LADAR [23], ultrasonic range finders [12], and Global Positioning System (GPS) receivers [2, 11]. The primary sensor of interest for this quadrotor platform is a scanning LADAR capable of providing range measurements in a plane from 0 degrees to 360 degrees.

This thesis is a direct extension of the work by Michael Stepaniak at Ohio University [23].

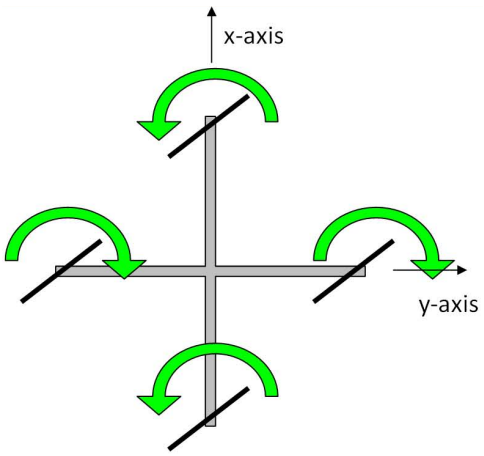

Figure 1. Quadrotor Viewed From Above With Z-Axis Directed Down Into The Page.

2.2 Quadrotor Operation

The typical quadrotor can be described as having a lightweight frame with four rotors affixed some distance away from the center of the frame, usually in a symmetric diamond or square pattern similar to that shown in Figure 1. Rotors opposite each other rotate in the same direction while adjacent rotors rotate opposite each other. The quadrotor achieves lift using a combination of thrust from each of the four rotors. Hovering is achieved when each rotor supplies thrust equal to one-fourth the weight of the vehicle. The quadrotor can pitch and roll about the x- and y-axes by decreasing thrust on one rotor and increasing the thrust on the rotor opposite as shown in Figures 2 and 3. Yawing is achieved when the thrust on an opposite pair of rotors is decreased and the thrust on the other rotors is increased so that the overall thrust remains constant, but the net torque about the quadrotor's z-axis is increased as shown in Figure 4. In these figures, blue enlarged arrows indicate an increased in rotational rate while red diminished arrows indicate a decreased rotational rate.

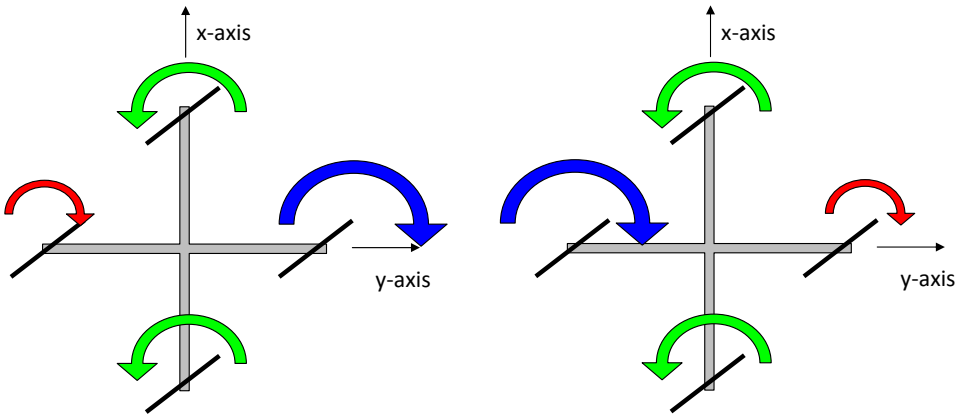

Figure 2. Quadrotor Rolling.

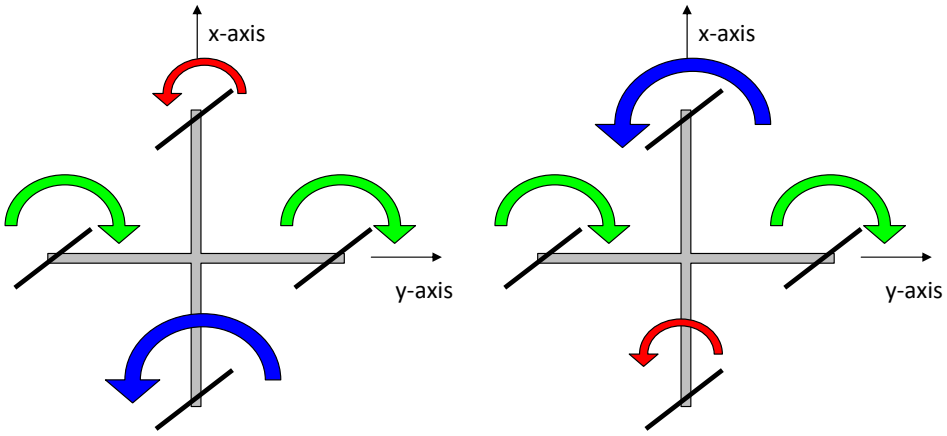

Figure 3. Quadrotor Pitching.

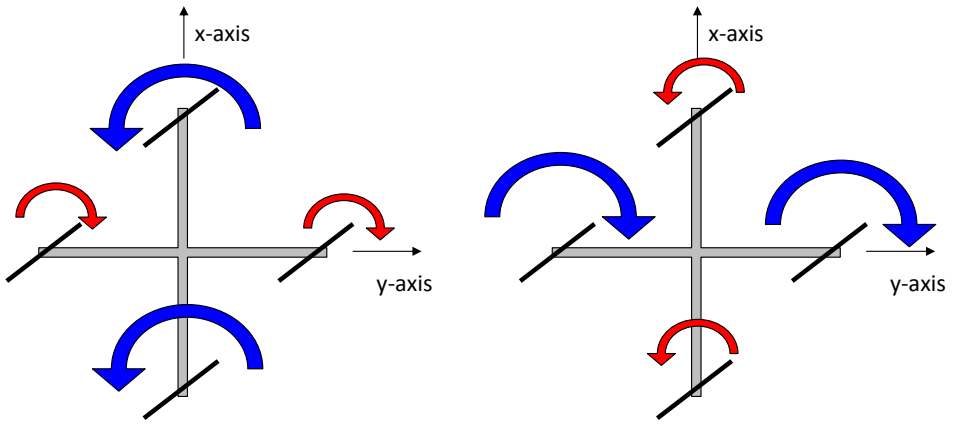

Figure 4. Quadrotor Yawing.

2.3 Equations Governing Motion and Measurements

2.3.1 Equations of Motion.

This section briefly presents the equations of motion. A more detailed presentation can be found in Reference [23]. The forces acting on a body which is both accelerating and rotating such as the quadrotor does in flight are described by:

$$\begin{aligned} F_x^b &= m(\dot{u}^b + qw^b - rv^b) \\ F_y^b &= m(\dot{v}^b + ru^b - pw^b) \\ F_z^b &= m(\dot{w}^b + pv^b - qu^b) \end{aligned} \tag{1}$$

where the x, y, and z components of the force are described in terms of mass of the quadrotor; the x-, y-, and z-axis accelerations ($\dot{u}$, $\dot{v}$, and $\dot{w}$ respectively); x-, y-, and z-axis velocities (u, v, and w respectively); and x-, y-, and z-axis angular rates (p, q, and r respectively).

The moments acting on the quadrotor are descibed by:

$$\begin{aligned} \sum L &= I_x\dot{p} - I_{xy}\dot{q} - I_{xz}\dot{r} - I_{xz}pq + I_{xy}pr + (I_z - I_y)qr + I_{yz}(r^2 - q^2) \\ \sum M &= I_y\dot{q} - I_{xy}\dot{p} - I_{yz}\dot{r} - I_{xy}qr + I_{yz}pq + (I_x - I_z)pr + I_{xz}(p^2 - r^2) \\ \sum N &= I_z\dot{r} - I_{xz}\dot{p} - I_{yz}\dot{q} - I_{yz}pr + I_{xz}qr + (I_y - I_x)pq + I_{xy}(q^2 - p^2) \end{aligned} \tag{2}$$

where L, M, and N represent the moments about the x-, y-, and z-axes respectively and the I terms represent the moments of inertia about the x-, y-, and z-axes as well as the xy-, xz-, and yz-axes. The $\dot{p}$, $\dot{q}$, and $\dot{r}$ terms represent the x-, y-, and z-axis angular accelerations respectively. These equations can be simplified using the assumptions that the quadrotor is symmetric about the x-z and y-z planes and that the x- and y-axis moments of inertia are nearly equal. From the table in section 3.1, these assumptions are valid since the cross moments of inertia are several orders of

magnitude less than on-axis moments of inertia. The resulting simplified equations are:

$$\begin{aligned} \sum L &= I_x\dot{p} + (I_z - I_y)qr \\ \sum M &= I_y\dot{q} + (I_x - I_z)pr \\ \sum N &= I_z\dot{r} \end{aligned} \tag{3}$$

2.3.2 Measurement Equations.

The primary interest in this section is what information can be gleaned from the accelerometers and gyro measurements. The IMU used on this quadrotor has three accelerometers, three rate gyros, and three magnetometers aligned to the three body axes of the quadrotor; however, the magnetometers are not used. The rate gyros directly measure the quadrotor's angular rates (p, q, and r) in radians per second and the accelerometers measure specific force on the quadrotor in g's. The gyros will be examined first, followed by an examination of the accelerometers.

2.3.2.1 Gyros.

There are two primary ways in which the quadrotor's attitude can be determined. The first, and more accurate way, is by solving the system of differential equations:

$$\begin{aligned} \dot{\phi} &= (q\sin(\phi) + r\cos(\phi))\tan(\theta) + p \\ \dot{\theta} &= q\cos(\phi) - r\sin(\phi) \\ \dot{\psi} &= (q\sin(\phi) + r\cos(\phi))\sec(\theta) \end{aligned} \tag{4}$$

where $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$ are the Euler angle rates and ϕ, θ, and ψ are the Euler angles [26]. The second, and the one that is implemented on the quadrotor, is extracting the Euler angles from the direction cosine matrix (DCM) which is propagated through time using the gyro measurements. The DCM is simply a rotation matrix which stores information about the quadrotor's attitude relative to the navigation frame. The

DCM which describes the rotation from the quadrotor's body frame to the navigation frame is:

$$C_b^n = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\theta\sin\phi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\theta\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \quad (5)$$

[23, 26]. The DCM describing the rotation from the navigation frame to the body frame is the transpose of the previous DCM and is:

$$C_n^b = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\theta\sin\phi & \sin\phi\sin\theta\sin\psi + \cos\theta\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \quad (6)$$

[23, 26]. The subscripts and superscripts b and n represent the body and navigation frame respectively and the direction of the DCM transformation can be read as from the subscript frame to the superscript frame.

The DCM is propagated through time by:

$$\dot{C}_b^n = C_b^n(t) \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \quad (7)$$

[23, 26] and the discrete time approximation used on the quadrotor is:

$$C_b^n(k+1) \approx C_b^n(k) \left(I + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \Delta t_m \right) \quad (8)$$

[23, 26]. The Δt_m term is the time step between the DCM at time k and the DCM

at time k+1 and is also the IMU sampling period of 0.01 seconds.

2.3.2.2 Accelerometers.

According to Titterton and Weston, "An accelerometer is insensitive to the gravitational acceleration (g) and thus provides an output proportional to the non-gravitational force per unit mass (f) to which the sensor is subjected along its sensitive axis" [26]. As a result, an accelerometer configured in the North-East-Down frame placed on a flat surface will not measure zero g's as might be expected since the accelerometer is not moving, nor does it measure one positive g since the acceleration due to gravity is along the positive z-axis of the accelerometer. Instead, the accelerometer measures one negative g, which is the normal force of the table acting on the accelerometer along the negative z-axis as shown:

$$\vec{a}^{b}_{meas} = \begin{bmatrix} 0 \\ 0 \\ -\frac{F_{thrust}}{m} \end{bmatrix} \tag{9}$$

So, the accelerometers mounted to the quadrotor will only measure accelerations due to thrust and aerodynamic drag. Since aerodynamic drag is assumed to be negligible and ignored in this thesis, the accelerometers only measure the acceleration due to thrust along the negative z^b-axis as shown: where $\vec{a}^{b}_{meas}$ is the accelerometer measurement vector and F_{thrust} is the total thrust.

Another way of thinking about this is to examine what would happen if the quadrotor were to pitch down. One might think that in this situation, the x^b-axis accelerometer might measure a portion of the gravity vector. However, the acceleration measured here is equal and opposite to the acceleration the accelerometer experiences due to the portion of the thrust of the quadrotor acting in the x^n-axis. It is a common

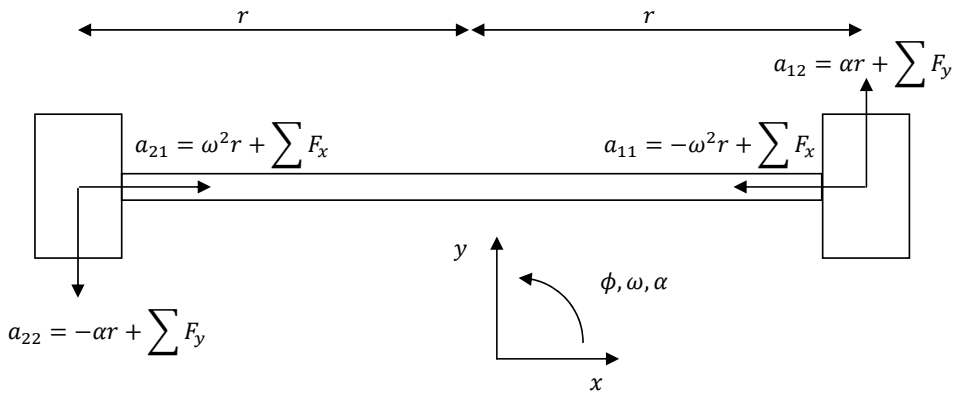

Figure 5. 2-Dimensional Quadrotor (Bi-rotor) With X and Y axis Accelerometers at Each End.

misconception to assume that what works on a test stand using accelerometers will work in flight. Many researches have found that they were able to use accelerometers to measure the gravity vector, or rather the normal force opposing the gravity vector, while on a test stand, only to discover that it did not work in free flight. In fact, I made the same mistake at the beginning of my work and quickly learned it would not work when I tried to simulate it.

If, however, aerodynamic effects are included, the accelerometers will be able to measure forces due to aerodynamic effects. The accelerometers readings can be integrated to determine velocity and the velocity along the x^b- and y^b-axes is proportional to the pitch and roll angles [15]. In the absolute best case, the quadrotor would be moving at constant velocity when the acceleration due to thrust is perfectly balanced by weight and aerodynamic drag forces. In that case, the accelerometers would in fact measure the force opposing the gravity vector directly allowing easy determination of the pitch and roll angles. This will be shown in Section 5.5.

Accelerometers can be used to measure angular rates and accelerations if a number of single axis accelerometers are placed around the quadrotor in patterns suggested in [4, 24]. A very simple error analysis is performed in Section 5.6 based on the simplified 2-dimensional example shown in Figure 5.

2.3.3 Forces and Moments.

Assuming no aerodynamic forces, the only forces acting on the quadrotor are those produced by the rotors. The rotors also produce the only torques acting on the quadrotor. The different effects produced by varying the rotor's speeds was already discussed in Section 2.2. The forces acting on the quadrotor in the body frame are:

$$\vec{F}^b = -\begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix} + mC_n^b \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{10}$$

and the moments acting on the quadrotor are:

$$\begin{aligned} \textstyle\sum L &= (F_4 - F_2)d \\ \textstyle\sum M &= (F_1 - F_3)d \\ \textstyle\sum N &= \tau_1 - \tau_2 + \tau_3 - \tau_4 \end{aligned} \tag{11}$$

The forces F_i describe the forces generated by the four motors and the torques τ_i describes the torques generated by the four rotors. The constant d is the distance from the quadrotor's center to the rotor's center, the length of the lever arm along which the force due to the rotors act to create moments about the x^b- and y^b-axes. The $\vec{F}^b$ is the total force vector in the body frame.

Eqaution 11 can now be combined with Equation 3 to form:

$$\begin{aligned} (F_4 - F_2)d &= I_x\dot{p} + (I_z - I_y)qr \\ (F_1 - F_3)d &= I_y\dot{q} + (I_x - I_z)pr \\ \tau_1 - \tau_2 + \tau_3 - \tau_4 &= I_z\dot{r} \end{aligned} \tag{12}$$

To make these equations easier to work with, they are linearized about a hover as

shown:

$$\begin{aligned}(F_4 - F_2)d &= I_x\dot{P} \\ (F_1 - F_3)d &= I_y\dot{Q} \\ \tau_1 - \tau_2 + \tau_3 - \tau_4 &= I_z\dot{R}\end{aligned} \tag{13}$$

and rewritten in state-space form as:

$$\begin{bmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix} = \begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix} + \begin{bmatrix}0 & -\frac{d}{I_x} & 0 & \frac{d}{I_x}\\ \frac{d}{I_y} & 0 & -\frac{d}{I_x} & 0\\ \frac{1}{I_z} & -\frac{1}{I_z} & \frac{1}{I_z} & -\frac{1}{I_z}\end{bmatrix}\begin{bmatrix}u_1\\ u_2\\ u_3\\ u_4\end{bmatrix} \tag{14}$$

Also note that rather than representing the states as delta states, they are represented as the true states since in a hover, the euler angles and angular rates are all zero and perturbations from that are equivalent to the true states. This is not true for the forces and torques since they will have some non-zero nominal value, but they are not represented as deltas for simplicity. Also, note that the forces and torques have been replaced by inputs u_i terms since the thrust and torque for the i^{th} rotor is set by the counts signal from the FPGA which is described in Sections 2.5 and 3.2.

However, the inputs are not in a very useful form since it is more desirable to control the Euler angles (or their rates) and the thrust. To achieve this, the following relationship:

$$\begin{bmatrix}u_p\\ u_q\\ u_r\\ u_{thrust}\end{bmatrix} = \begin{bmatrix}0 & -1 & 0 & 1\\ 1 & 0 & -1 & 0\\ 1 & -1 & 1 & -1\\ 1 & 1 & 1 & 1\end{bmatrix}\begin{bmatrix}u_1\\ u_2\\ u_3\\ u_4\end{bmatrix} \tag{15}$$

is used and then applied to Equation 14 resulting in:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{d}{I_x} & 0 & 0 & 0 \\ 0 & \frac{d}{I_y} & 0 & 0 \\ 0 & 0 & \frac{1}{I_z} & 0 \end{bmatrix} \begin{bmatrix} u_p \\ u_q \\ u_r \\ u_{thrust} \end{bmatrix} \tag{16}$$

The matrix shown here is referred to as a mixer matrix. It is directly derived from Equations 10 and 11. The commanded roll is equal to the difference between the fourth and second motor commands. The commanded pitch is equal to the difference between the first and third motor commands. The commanded yaw is equal to the difference between the sum of the odd motor commands and the sum of the even motor commands. The commanded thrust is equal to the sum of all four motor commands.

Now, the state space representation is converted to a transfer function in:

$$\begin{bmatrix} p(s) \\ q(s) \\ r(s) \end{bmatrix} = \begin{bmatrix} \frac{d}{I_x s} & 0 & 0 & 0 \\ 0 & \frac{d}{I_y s} & 0 & 0 \\ 0 & 0 & \frac{1}{I_z s} & 0 \end{bmatrix} \begin{bmatrix} u_p(s) \\ u_q(s) \\ u_r(s) \\ u_{thrust}(s) \end{bmatrix} \tag{17}$$

Note that this is not the plant model for the quadrotor yet since the characteristics of the motor and the controller sampling rate have not yet been included.

After the motor and sampling transfer functions are included, the plant models

for the roll, pitch, and yaw channels are:

$$\begin{aligned} u_p(s) &= \frac{d}{I_x s}\frac{4.172\times10^{-4}\times24.5}{s+24.5}\frac{100}{s+100} \\ u_q(s) &= \frac{d}{I_y s}\frac{4.172\times10^{-4}\times24.5}{s+24.5}\frac{100}{s+100} \\ u_r(s) &= \frac{1}{I_z s}\frac{4.172\times10^{-4}\times24.5}{s+24.5}\frac{100}{s+100} \end{aligned} \tag{18}$$

[23] where the 4.172×10^{-4} term is a result of linearizing the motor thrust curve in Equation 25 about a hover.

2.4 Control Scheme

The controller's used are the same as those used in Reference [23] and are depicted graphically in Figures 7, 8, and 9. These controller's were verified to work with the non-linear model presented in Section 4.1 and that verification is shown graphically in Figure 6 where roll, pitch, and yaw step inputs were applied and noiseless measurements were fed back to the control loop. The roll, pitch, and yaw controllers show that the quadrotor response to a step input reaches steady state in one to two seconds. Yaw responds to commands in roll and pitch since there is coupling between roll, pitch, and yaw in the non-linear equations and yaw angle is not being fed back. Pitch and roll controllers both make use of a lead compensator within the inner loop as shown:

$$H_{lead}(s) = \frac{9.25(s+7.5)}{s+69.4} \tag{19}$$

The inner loop of these controllers makes use of rate feedback while the outer loop makes use of position feedback. The inner loop gain is 2^{11} and the outer loop gain is 2^3. The gains were chosen to be powers of 2 so that they would be easy to implement on an FPGA as a bit shift.

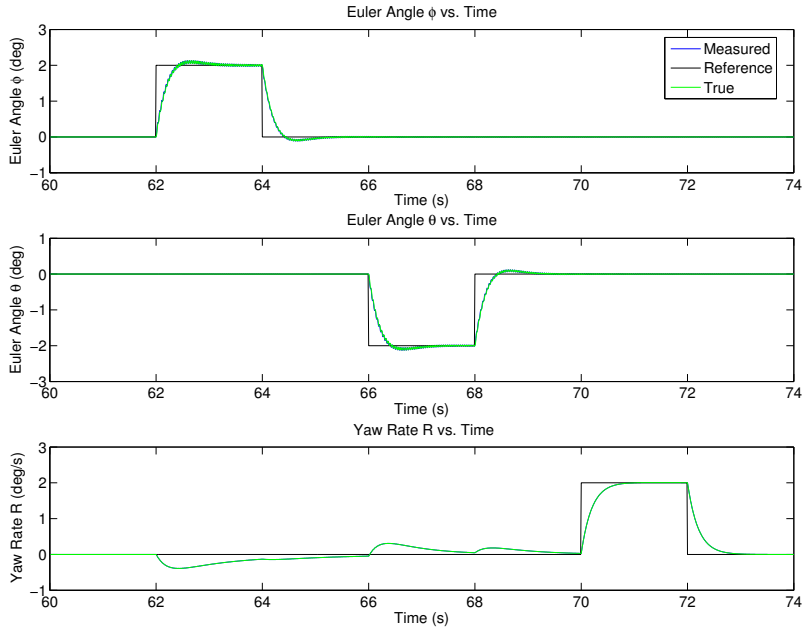

Figure 6. Controller Verification.

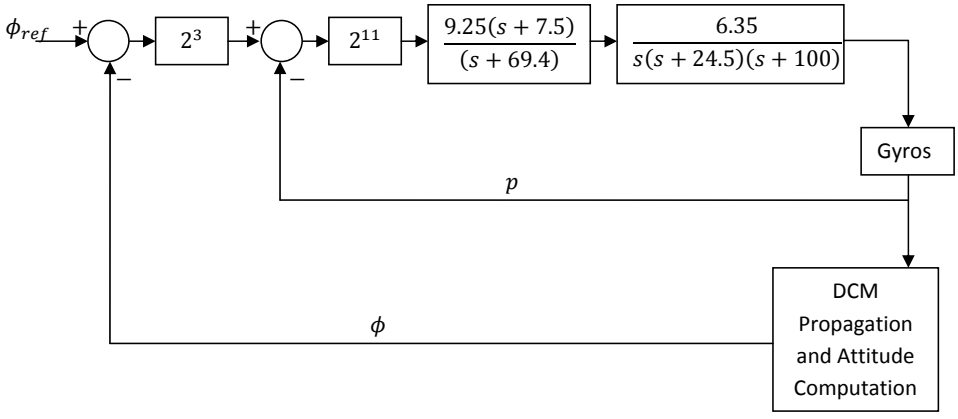

Figure 7. Roll controller surrounding the quadrotor roll channel plant model.

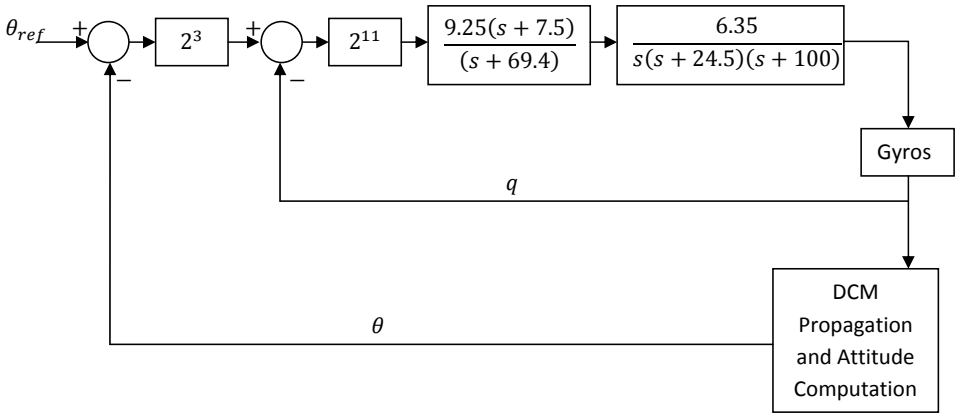

Figure 8. Pitch controller surrounding the quadrotor pitch channel plant model.

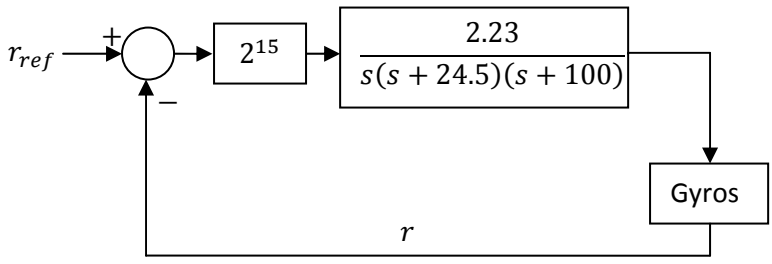

Figure 9. Yaw controller surrounding the quadrotor yaw channel plant model.

2.5 Controller Implementation

At this point, it is useful to look at how the controller is implemented on board the FPGA. For the roll and pitch controllers, the reference input and feedback to the first difference block are in radians. The gain following the first difference block also converts the units to radians per second before subtracting the angular rate in the second difference block. The second gain block converts from radians per second to FPGA counts (a unitless measure based on the 50MHz FPGA clock rate) before passing through the lead compensator. Similarly, for the yaw controller, the reference input and feedback to the first difference block are in radians per second. The first gain block converts from radians per second to FPGA counts. The thrust controller is still in terms of pounds-force even after the division block. To convert the thrust to counts, the output of the division block is divided by 4 since the thrust curve in Equation 25 only works for one motor at a time, and the thrust curve equation is then solved for PWM command. The PWM command is then converted to counts according to Equation 28 and multiplied by 4 so that the command will be divided between the motors after passing through the inverse of the mixer matrix in Equation 15. These values are then converted to individual motor commands in FPGA counts through use of the mixer matrix in Equation 15 as shown:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} u_p \\ u_q \\ u_r \\ u_{thrust} \end{bmatrix} \tag{20}$$

III. Quadrotor Development

The following sections discuss how the quadrotor is configured structurally and electronically. The structural configuration section discusses the physical design and layout of all physical components of the quadrotor along with the mass and moment of inertia properties of the entire quadrotor. The electronics configuration section discusses the power systems, the hardware used to implement the controller, and how the controller communicates with the radio receiver, inertial measurement unit, and motors. It is important to note that the quadrotor used in this thesis receives roll angle, pitch angle, yaw rate, and thrust commands from a human pilot using a radio link.

There are a few different motivations for creating a new structural design for this quadrotor. The first was a desire for a smaller vehicle which can more easily be operated indoors and carried around indoors while in transport. Another was the desire to more efficiently cool the motor batteries. In the previous design, the motor batteries were all clumped together in the center of the quadrotor and isolated from air flow causing them to heat up significantly. The new design places the batteries away from each other and allows air flow from the rotors to cool them. Finally, since the primary sensor of interest is a scanning laser detection and ranging (LADAR) sensor which weighs approximately 6 pounds, there was a desire to create a more rigid attachment point.

3.1 Structural Configuration

To begin this section it is important to note the assumptions made in computing the quadrotor's moments of inertia. It is assumed that all quadrotor components are uniformly dense with the center of mass coinciding with the center of volume. Each

component is assumed to be a rectangular solid. These are the same assumptions made in reference [23]. The location of each of the components in relation to the quadrotor's assumed center of the gravity is specified by the location of the component's center of mass and relative rotation to the body frame. A summary of these properties are shown in Tables 1 and 2. The structural comonents listed as booms, plates, and battery supports are made of aluminum. The booms are rectangular prisms, 0.75 inches on a side and hollow inside with a inner to outer wall thickness of 0.125 inches. The battery supports are right angle, U-channel pieces of aluminum with inner to outer wall thickness of 0.125 inches. The interior depth of the U-channel is 1.125 inches and the interior width of the U-channel is 1 inch.

Table 1. Component Mass, Size, and Position Properties.

Component	Center of Object from Center of Vehicle			Dimensions			Orientation			Mass(lbs)
	x(ft)	y(ft)	z(ft)	x(ft)	y(ft)	z(ft)	ϕ(deg)	θ(deg)	ψ(deg)	
Boom 1	0	0	0	2.0807	0.0625	0.0625	0	0	0	0.7125
Boom 2	0	0.5352	0	1.0078	0.0625	0.0625	0	0	90	0.3250
Boom 3	0	-0.5339	0	1.0052	0.0625	0.0625	0	0	90	0.3375
Plate 1	0	0	-0.0339	0.4974	0.5	0.0052	0	0	0	0.2125
Plate 2	0	0	0.0339	0.4974	0.4974	0.0052	0	0	0	0.2125
Battery Support 1	0.4425	0.4425	-0.0104	1.3281	0.1042	0.1042	0	0	-45	0.3625
Battery Support 2	-0.4425	0.4425	-0.0104	1.3255	0.1042	0.1042	0	0	45	0.3625
Battery Support 3	-0.4425	-0.4425	-0.0104	1.3281	0.1042	0.1042	0	0	-45	0.3625
Battery Support 4	0.4425	-0.4425	-0.0104	1.3255	0.1042	0.1042	0	0	45	0.3625
Battery 1	0.4425	0.4425	-0.0771	0.8073	0.0833	0.2167	0	0	-45	1.8000
Battery 2	-0.4425	0.4425	-0.0771	0.8073	0.0833	0.2167	0	0	45	1.8000
Battery 3	-0.4425	-0.4425	-0.0771	0.8073	0.0833	0.2167	0	0	-45	1.8000
Battery 4	0.4425	-0.4425	-0.0771	0.8073	0.0833	0.2167	0	0	45	1.8000
Motor 1	0.9271	0	-0.1536	0.0833	0.0833	0.1823	0	0	0	0.6875
Motor 2	0	0.9271	-0.1536	0.0833	0.0833	0.1823	0	0	0	0.6875
Motor 3	-0.9271	0	-0.1536	0.0833	0.0833	0.1823	0	0	0	0.6875
Motor 4	0	-0.9271	-0.1536	0.0833	0.0833	0.1823	0	0	0	0.6875
Rotor 1	0.9271	0	-0.3385	1	1	0.125	0	0	0	0.1875
Rotor 2	0	0.9271	-0.3385	1	1	0.125	0	0	0	0.1875
Rotor 3	-0.9271	0	-0.3385	1	1	0.125	0	0	0	0.1875
Rotor 4	0	-0.9271	-0.3385	1	1	0.125	0	0	0	0.1875

Table 2. Component Mass, Size, and Position Properties Continued.

Component	Center of Object from Center of Vehicle			Dimensions			Orientation			Mass(lbs)
	x(ft)	y(ft)	z(ft)	x(ft)	y(ft)	z(ft)	ϕ(deg)	θ(deg)	ψ(deg)	
ESC 1	0.5	0.0479	0	0.2042	0.0833	0.0333	90	0	0	0.1250
ESC 2	-0.0479	0.5	0	0.2042	0.0833	0.0333	90	0	90	0.1250
ESC 3	-0.5	-0.0479	0	0.2042	0.0833	0.0333	90	0	0	0.1250
ESC 4	0.0479	-0.5	0	0.2042	0.0833	0.0333	90	0	90	0.1250
Avionics Battery	0.1146	-0.0833	0	0.1667	0.1042	0.0417	0	0	0	0.1
Voltage Regulator	0.1146	-0.1771	0	0.1458	0.0833	0.0417	0	0	0	0.03
FPGA	-0.0833	0	-0.0698	0.3333	0.5	0.0667	0	0	0	0.2375
IMU	0.1563	0.0938	0.0125	0.25	0.125	0.0375	0	0	0	0.05
Radio Receiver	0.1667	-0.1667	-0.0677	0.125	0.1875	0.0625	0	0	0	0.07
LD-OEM LADAR	0	0	-0.4840	0.3950	0.3767	0.8950	0	0	0	5.4

The quadrotor's center of gravity was computed by essentially integrating the density of the various components over their volumes at the exact location they would be in the quadrotor. This was done in Matlab by creating a "cloud" of uniformly distributed points to represent each component such that each point in the cloud had an equal fraction of the total mass of that component. These point masses were summed along x^b-, y^b-, and z^b-axes (the b superscript represents the body frame axes) of the quadrotor according to:

$$\begin{aligned} CG_x &= \tfrac{1}{m}\textstyle\sum_{i=1}^{n} m_i x_i^b \\ CG_y &= \tfrac{1}{m}\textstyle\sum_{i=1}^{n} m_i y_i^b \\ CG_z &= \tfrac{1}{m}\textstyle\sum_{i=1}^{n} m_i z_i^b \end{aligned} \tag{21}$$

to determine the quadrotor's center of gravity relative to its initially assumed center of gravity. In these equations, m represents the total mass of the quadrotor, m_i represents the mass of the i^{th} point mass, x_i^b, y_i^b, and z_i^b represent the location of the i^{th} point mass relative to the assumed center of gravity where i is an integer, and n represents the total number of point masses in the quadrotor. The result of these computations is:

$$\begin{array}{cc} \text{Without LADAR Payload} & \text{With LADAR Payload} \\ CG_x = 9.773 \times 10^{-4}(\text{ft}) & CG_x = 7.171 \times 10^{-4}(\text{ft}) \\ CG_y = -1.799 \times 10^{-3}(\text{ft}) & CG_y = -1.321 \times 10^{-3}(\text{ft}) \\ CG_z = -8.483 \times 10^{-2}(\text{ft}) & CG_z = -1.908 \times 10^{-1}(\text{ft}) \end{array} \tag{22}$$

where CG_x, CG_y, and CG_z are the x, y, and z center of gravity offsets respectively.

These point masses were used again to compute the moments of inertia where the usual integrals were replaced by summations. The equations used to compute the

moments of inertia are:

$$\begin{aligned} I_x &= \textstyle\sum_{i=1}^{n} m_i((y_i^b)^2 + (z_i^b)^2) \quad I_{xy} = \sum_{i=1}^{n} m_i(x_i^b y_i^b) \\ I_y &= \textstyle\sum_{i=1}^{n} m_i((x_i^b)^2 + (z_i^b)^2) \quad I_{xz} = \sum_{i=1}^{n} m_i(x_i^b z_i^b) \\ I_z &= \textstyle\sum_{i=1}^{n} m_i((x_i^b)^2 + (y_i^b)^2) \quad I_{yz} = \sum_{i=1}^{n} m_i(y_i^b z_i^b) \end{aligned} \tag{23}$$

and the results of those computations are:

$$\begin{aligned} &\text{Without LADAR Payload} \\ I_x &= 1.285 \times 10^{-1}(\text{slug} \cdot \text{ft}^2) \quad I_{xy} = -9.267 \times 10^{-5}(\text{slug} \cdot \text{ft}^2) \\ I_y &= 1.289 \times 10^{-1}(\text{slug} \cdot \text{ft}^2) \quad I_{xz} = 2.139 \times 10^{-5}(\text{slug} \cdot \text{ft}^2) \\ I_z &= 2.426 \times 10^{-1}(\text{slug} \cdot \text{ft}^2) \quad I_{yz} = 2.635 \times 10^{-5}(\text{slug} \cdot \text{ft}^2) \\ &\text{With LADAR Payload} \\ I_x &= 1.813 \times 10^{-1}(\text{slug} \cdot \text{ft}^2) \quad I_{xy} = -9.268 \times 10^{-5}(\text{slug} \cdot \text{ft}^2) \\ I_y &= 1.819 \times 10^{-1}(\text{slug} \cdot \text{ft}^2) \quad I_{xz} = 2.140 \times 10^{-5}(\text{slug} \cdot \text{ft}^2) \\ I_z &= 2.470 \times 10^{-1}(\text{slug} \cdot \text{ft}^2) \quad I_{yz} = 2.635 \times 10^{-5}(\text{slug} \cdot \text{ft}^2) \end{aligned} \tag{24}$$

where I_x, I_y, I_z, I_{xy}, I_{xz}, and I_{yz} are the x-, y-, z-, xy-, xz-, and yz-axis moments of inertia respectively.

An example of what that point cloud looks like when each component is made up of approximately 100 points is shown in Figure 10. The number of points per component used to compute the center of gravity and moments of inertia was approximately 100,000. The colors in the figure were used to help identify specific components. The large red bricks represent the batteries, electronic speed controllers, and voltage regulator. The green bricks represent the motors. The yellow bricks represent the rotors. The black represents the structure. The magenta represents the radio receiver and IMU. The cyan represents the FPGA. Of note is the placement of the batteries relative to where they were placed on Ohio University's quadrotor. Due to overheating

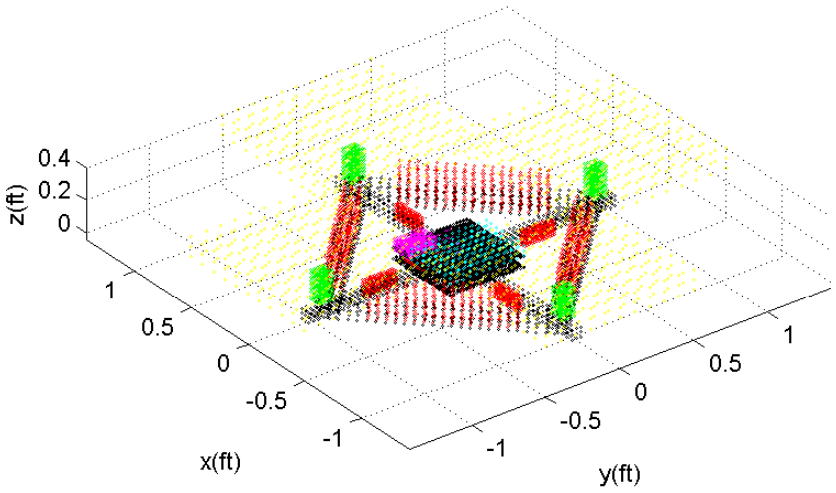

Figure 10. Quadrotor Represented As A Point Cloud To Compute Center Of Gravity And Moments Of Inertia.

issues, the batteries were moved out of the central body area and located between the boom arms so the rotors can aid in cooling the batteries. The difference can be visualized by comparing the new quadrotor configuration relative to the previous quadrotor configuration shown as a point cloud in Figure 11 where the same color scheme is used. The only difference is that the previous configuration uses blue to denote both the rotors and motors as individual objects rather than separate objects as in the image of the new configuration. Also of note is that a smaller central plate is used and the motors have been brought a little closer to the center of gravity. The smaller, more compact design allows for the quadrotor to be more easily used indoors. The smaller central plate allows a rigid attachment point for the scanning LADAR.

Towards the end of this thesis, a Lantronix WiPort Evaluation Board was added in order to wirelessly transmit data from the FPGA to a laptop acting as a ground station for testing purposes. This was facilitated by mounting the WiPort on a small

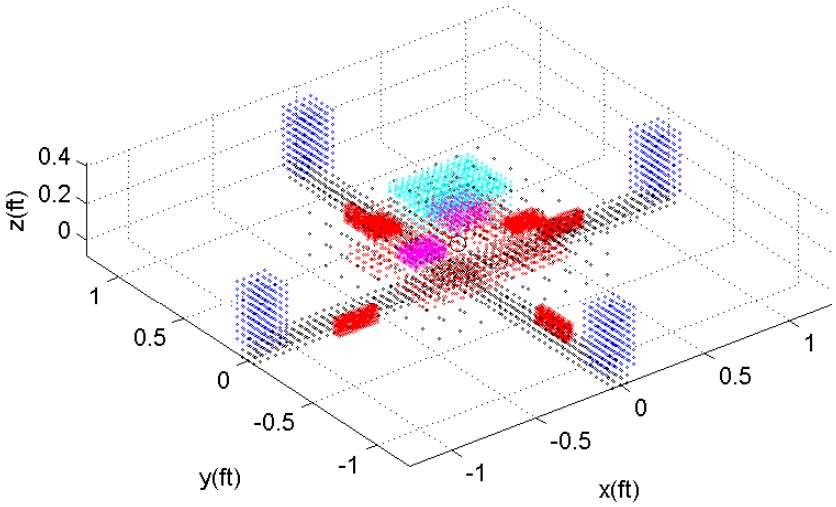

Figure 11. Previous Quadrotor Represented As A Point Cloud.

piece of the same material used for the top and bottom plates. The WiPort plate was then attached to the top plate using a piece of U-channel. The WiPort also required a separate battery and voltage regulator. The addition of the WiPort, battery, voltage regulator, and mount for the WiPort, were not factored into the calculation of the moments of inertia or CG or into the simulations.

3.2 Electronics Configuration

To begin, the on board processing for the quadrotor is performed by a one million gate Spartan-3 field programmable gate array (FPGA) that operates at 50MHz. The FPGA receives instructions from the remote pilot using a Futaba T6EXAP remote control which sends commands to a Futaba R156F receiver on board the quadrotor at a rate of 50Hz. Originally, a Spektrum DX7 remote control and Spektrum AR7000 receiver were to be used, but the receiver would periodically stop transmitting signals

to the FPGA. The FPGA additionally receives 3-axis acceleration and angular rate data from a Microstrain 3DM-GX3-25 IMU at a rate of 100Hz over a serial cable. The FPGA commands the motors by sending commands to an electronic speed controller (ESC) for each motor. The ESCs are Castle Creations Phoenix-60 brushless motor controllers. The ESCs convert these commands from digital command signals to pulse width modulation (PWM) commands which then go to the motors. Each motor is powered by a Thunder Power 18.5V, 8000mAh, 5 cell Lithium-Polymer (LiPo) battery which is connected to the ESC which is then connected to the motors. The motors are AXi 4120/18 brushless motors. The FPGA, IMU, and AR7000 are powered by a separate Thunder Power 7.4V, 910mAh, 2 cell LiPo battery. The AR7000 is powered directly from the battery while the FPGA and IMU receive regulated 5V from a Castle Creations Battery Eliminator Circuit (BEC).

The addition of the WiPort required the addition of another power supply, a Thunder Power 11.1V, 1350mAh, 3 cell LiPo battery connected to a Dimension Engineering DE-SW033 3.3V regulator. The data from the FPGA is sent to the WiPort over the FPGA's secondary serial connection.

A diagram of the electronics is shown if Figure 12. The arm command in the figure allows the pilot to enable/disable the motors remotely for safety. The motor thrust curve was previously determined as:

$$F_T = 2.02 \times 10^{-5}\text{PWM}^2 + 6.39 \times 10^{-3}\text{PWM} - 0.8 \tag{25}$$

[23] where the input is PWM commands from the ESC and F_T is the thrust produced.

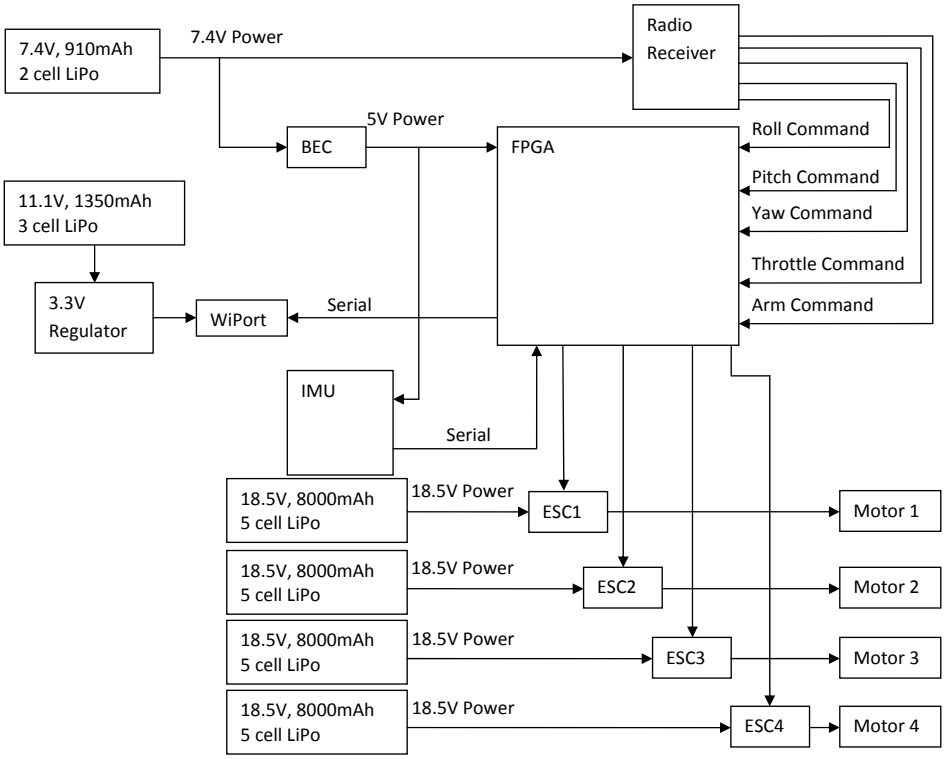

Figure 12. Electronics Configuration.

The motor torque is related to the thrust by:

$$\begin{aligned} C_T &= \frac{F_T}{\rho \pi R^4 \omega^2} = 1.568 \times 10^{-2} \\ C_Q &= \frac{\tau}{\rho \pi R^5 \omega^2} = 2.359 \times 10^{-3} \\ \tau &= \frac{C_Q}{C_T} R_l F_T = 0.0878 F_T \end{aligned} \tag{26}$$

where C_T is the thrust coefficient, C_Q is the torque coefficient, τ is the torque, ρ is the air density, R_l is the rotor blade radius, and ω is the rotor angular speed. The rotor blade radius is 7 inches, but is converted to feet for the computation. Additionally, the motor has a first order lag described by the transfer function:

$$H_{motor}(s) = \frac{24.5}{s + 24.5} \tag{27}$$

where $H_{motor}(s)$ is the motor transfer function. Another important conversion factor

is from the commands issued by the FPGA, counts, to the PWM commands issued by the ESCs. That conversion is:

$$\text{PWM} = 0.02\text{counts} \tag{28}$$

Finally, since the control loop operates at 50Hz, a first order lag approximation based on the Padé approximation is given by the transfer function:

$$H_{sampler}(s) = \frac{100}{s + 100} \tag{29}$$

where $H_{sampler}(s)$ is the sampler transfer function.

IV. Methodology

4.1 Development

An accurate non-linear model is important for this thesis since it will allow comparison of how the quadrotor flies in simulation against how it will fly in reality. Since the desire for this project is to navigate in indoor, urban, and underground environments, it is assumed that the velocities at which the quadrotor will fly are sufficiently low enough to neglect aerodynamic effects on the quadrotor body. The aerodynamic effects of the rotors themselves is also ignored for the sake of simplicity. The non-linear model is created from Equations 1, 2, 10, and 11 and are repeated here:

$$\begin{aligned} F_x^b &= m(\dot{u}^b + qw^b - rv^b) \\ F_y^b &= m(\dot{v}^b + ru^b - pw^b) \\ F_z^b &= m(\dot{w}^b + pv^b - qu^b) \end{aligned} \tag{30}$$

$$\begin{aligned} \sum L &= I_x\dot{p} - I_{xy}\dot{q} - I_{xz}\dot{r} - I_{xz}pq + I_{xy}pr + (I_z - I_y)qr + I_{yz}(r^2 - q^2) \\ \sum M &= I_y\dot{q} - I_{xy}\dot{p} - I_{yz}\dot{r} - I_{xy}qr + I_{yz}pq + (I_x - I_z)pr + I_{xz}(p^2 - r^2) \\ \sum N &= I_z\dot{r} - I_{xz}\dot{p} - I_{yz}\dot{q} - I_{yz}pr + I_{xz}qr + (I_y - I_x)pq + I_{xy}(q^2 - p^2) \end{aligned} \tag{31}$$

$$\vec{F}^b = -\begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix} + mC_n^b \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{32}$$

$$\begin{aligned} \sum L &= (F_4 - F_2)d \\ \sum M &= (F_1 - F_3)d \\ \sum N &= \tau_1 - \tau_2 + \tau_3 - \tau_4 \end{aligned} \tag{33}$$

In order to simulate the non-linear model, Equation 30 is set equal to 32 and solved for $\dot{u}$, $\dot{v}$, and $\dot{w}$ and Equation 31 is set equal to 33 and solved for $\dot{p}$, $\dot{q}$, and

$\dot{r}$. At this point, a first order Taylor Series approximation [8] is used in order to determine the angular rates and accelerations at discrete times as shown:

$$\begin{aligned} u_{true}(k) &= u_{true}(k-1) + \dot{u}_{true}(k-1)\Delta t \\ v_{true}(k) &= v_{true}(k-1) + \dot{v}_{true}(k-1)\Delta t \\ w_{true}(k) &= w_{true}(k-1) + \dot{w}_{true}(k-1)\Delta t \end{aligned} \tag{34}$$

$$\begin{aligned} P_{true}(k) &= P_{true}(k-1) + \dot{P}_{true}(k-1)\Delta t \\ Q_{true}(k) &= Q_{true}(k-1) + \dot{Q}_{true}(k-1)\Delta t \\ R_{true}(k) &= R_{true}(k-1) + \dot{R}_{true}(k-1)\Delta t \end{aligned} \tag{35}$$

where k is a discrete time index. The time difference between k and $k-1$ is Δt which is the simulation time step, not to be confused with Δt_m. Note that true subscripts have been added to indicate that the non-linear model is assumed to be truth. Non-subscripted values will refer to measurements corrupted by noise. The solution for the accelerations and angular accelerations are computed in terms of the velocity, angular rates, forces, and torques at the previous time step. The forces and torques are computed according to Equations 25 and 26. The motor's first order lag in Equation 27 is included by taking the z-transform and propagating discretely as shown:

$$\begin{aligned} H_{motor}(z) &= \tfrac{a+bz^{-1}}{c+dz^{-1}} \\ F(k) &= \tfrac{-dF(k-1)+aF_{int}(k)+bF_{int}(k-1)}{c} \\ \tau(k) &= \tfrac{C_Q}{C_T} R_l F(k) \end{aligned} \tag{36}$$

where the a, b, c, d coefficients are place holders for their true values not shown here and F_{int} is the motor force computed according to Equation 25.

The pulse width modulation (PWM) inputs to Equation 25 are computed from the field programmable gate array (FPGA) counts according to Equation 28. The FPGA counts are determined as described in Section 2.5. The only thing not mentioned in

that section that is included here is how the lead compensator is incorporated into the simulation. It follows the same pattern as the motor transfer function and is implemented according to:

$$\begin{aligned} H_{lead}(z) &= \tfrac{a+bz^{-1}}{c+dz^{-1}} \\ u_p(k) &= \tfrac{-du_p(k-1)+a\phi_{intermediate}(k)+b\phi_{intermediate}(k-1)}{c} \\ u_q(k) &= \tfrac{-du_q(k-1)+a\theta_{intermediate}(k)+b\theta_{intermediate}(k-1)}{c} \end{aligned} \tag{37}$$

where ϕ_{int} and θ_{int} values are the FPGA counts at the output of the second gain block in the roll and pitch controllers. The inputs to the control loop are the measured values and not the true values.

The fidelity of the simulations increase as the value of Δt is decreased. Low fidelity simulations were performed with time steps 10 times smaller than the 100Hz inertial measurement unit (IMU) measurement rate while high fidelity simulations were performed with time steps 100 times smaller than the IMU measurement rate. Low fidelity simulations were performed to allow for quicker simulations in the earlier stages of developing the non-linear model and comparing it to the linear models developed in the next section. High fidelity simulations were used in reporting the results in Chapter IV. The true position is computed according to:

$$\begin{aligned} x_{true}(k) &= x_{true}(k-1) + u_{true}(k-1)\Delta t \\ y_{true}(k) &= y_{true}(k-1) + v_{true}(k-1)\Delta t \\ z_{true}(k) &= z_{true}(k-1) + w_{true}(k-1)\Delta t \end{aligned} \tag{38}$$

using a first order Taylor Series approximation [8]. The true Euler angles are com-

puted according to:

$$\begin{aligned}
\dot{\phi}_{true}(k-1) &= (q_{true}(k-1)\sin(\phi_{true}(k-1))+ \\
r_{true}(k-1)\cos(\phi_{true}(k-1)))\tan(\theta_{true}(k-1)) + p_{true}(k-1) \\
\dot{\theta}_{true}(k-1) &= q_{true}(k-1)\cos(\phi_{true}(k-1))- \\
& r_{true}(k-1)\sin(\phi_{true}(k-1)) \\
\dot{\psi}_{true}(k-1) &= (q_{true}(k-1)\sin(\phi_{true}(k-1))+ \\
r_{true}(k-1)\cos(\phi_{true}(k-1)))sec(\theta_{true}(k-1))
\end{aligned} \tag{39}$$

[26] and:

$$\begin{aligned}
\phi_{true}(k) &= \phi_{true}(k-1) + \dot{\phi}_{true}(k-1)\Delta t \\
\theta_{true}(k) &= \theta_{true}(k-1) + \dot{\theta}_{true}(k-1)\Delta t \\
\psi_{true}(k) &= \psi_{true}(k-1) + \dot{\psi}_{true}(k-1)\Delta t
\end{aligned} \tag{40}$$

using a first order Taylor Series approximation [8] as well.

At this point, the true accelerations and angular rates are sampled at 100Hz, the same rate as the IMU, and noise is added to those measurements. The exact nature of the noise is described in Section 4.3, but for now it will suffice to say that the measurement noise used can be either simulated, or real noise collected from stationary IMU data collects. However, due to noise modeling deficiencies, real noise was used in the simulations used in reporting results. Simulated estimates of the Euler angles are computed by first propagating the direction cosine matrix (DCM) discretely according to Equation 8 and the Euler angles are then estimated as:

$$\begin{aligned}
\phi &= C_b^n(3,2) \\
\theta &= -C_b^n(3,1) \\
\psi &= C_b^n(2,1)
\end{aligned} \tag{41}$$

where the small angle approximation has been used on the DCM presented in Equa-

tion 5.

4.2 Linear Model

First, the linear model will be developed and a method to validate it is proposed.

4.2.1 Development.

Since the roll, pitch, and yaw are being controlled, the states of interest in the linear model are the quadrotor's angular rates p, q, and r. These states can be integrated through discrete DCM propagation according to Equations 8 to determine the quadrotor's attitude according to 41. A state space model of the form:

$$\dot{\vec{x}} = A\vec{x} + B\vec{u} \tag{42}$$

is then developed by populating the A matrix according to:

$$A(i,j) = \frac{\partial \dot{\vec{x}}(i)}{\partial \vec{x}(j)}|_{nom} \quad i = 1,2,3 \quad j = 1,2,3 \tag{43}$$

and the B matrix according to:

$$B(i,j) = \frac{\partial \dot{\vec{x}}(i)}{\partial \vec{u}(j)}|_{nom} \quad i = 1,2,3 \quad u = 1,2,3,4 \tag{44}$$

The state vector and its derivative are defined as:

$$\vec{x} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad \dot{\vec{x}} = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \tag{45}$$

and the input vector is defined as:

$$\vec{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \tag{46}$$

where the subscript indicates the counts being sent from the FPGA to the electronic speed controller (ESC) for the first, second, third, and fourth motor respectively. The $\frac{\partial \dot{\vec{x}}(i)}{\partial \vec{x}(j)}|_{nom}$ term refers to the partial derivative of the i^{th} variable in the state vector derivative with respect to the j^{th} variable in the state vector evaluated at the linearization point. Similarly, the $\frac{\partial \dot{\vec{x}}(i)}{\partial \vec{u}(j)}|_{nom}$ term refers to the partial derivative of the i^{th} variable in the state vector derivative with respect to the j^{th} motor command evaluated at the linearization point. The linearization point is chosen to be a hover so that the roll, pitch, and yaw angles and rates and also position and velocity are all zero. This also sets the motor commands so that the vehicle can maintain a hover. As demonstrated in Section 2.3.3, the inputs can be converted to roll, pitch, yaw, and thrust commands through use of the mixer matrix. The linear model is simulated side by side with the non-linear model and has as inputs, the same commands that went to the motors based on measurements taken from the non-linear model minus the nominal inputs about which the linearization was computed.

The linear model is developed based on Equations 25, 26, 28, 31, and 33 where Equations 31 and 33 were set equal, the state vector derivative was solved for, and then used in the derivation of the A and B matrices according to Equations 43 and 44.

4.2.2 Validation.

The linear model is validated, or not, as the case may be, by comparing it with the non-linear model. If the linear model is capable of following the non-linear models at least as well as the drift error due to integrating the gyro measurements of the IMU to determine attitude, the linear model will be considered valid. Two different tests will be performed. Both will begin with a one minute period in which the bias will be calculated and removed from the acceleromter and angular rate measurements. The first test will command a 5 minute hover (static) following the bias removal period. The second test will command a roll command followed by a pitch command over a 10 second period (dynamic) following the bias removal period.

4.3 Measurement Model Development

Development of the measurement model begins with collecting data from the IMU in order to characterize the noise on the measurements and culminates with which measurements are used in controlling the quadrotor. Specifically, which measurements to be used comes down to a determination between whether to navigate off the raw measurements or to use state estimates generated by a Kalman Filter.

4.3.1 Noise Characterization.

In order to characterize the noise on the accelerometer and gyro readings from the IMU, 48 continuous hours of data was collected at the IMU sampling rate of 100Hz. This data is analyzed in two ways. The first being an Allan Variance analysis and the second being a power spectral density (PSD) analysis. The rationale for collecting 48 hours worth of data is so that both the Allan Variance analysis and PSD analysis can pick up on lower frequency errors such as long term measurement drift and oscillations.

There are numerous papers discussing how to perform an Allan Variance analysis on accelerometer and gyro measurements so the specific details are not discussed here [6, 13, 20, 27]. The Allan Variance analysis performed here was limited to only accounting for the random walk and bias instability terms for the sake of simplicity and because it was assumed that the higher order effects had little impact on the measurements for a typical quadrotor flight lasting less than 20 minutes.

The PSD analysis consists of computing the power spectral density of the accelerometer measurements, plotting that power spectral density in dB on a log scale, and determining poles and zeros for a transfer function that would approximate the PSD curve assuming white Gaussian noise as an input to that transfer function. The method of determining poles and zeros is identical to identifying poles and zeros on a Bode Plot, however, since the PSD is essentially the square of magnitude (as in a Bode plot), the slope for a single zero or pole are 40dB and -40dB rather than 20dB and -20dB.

A major consideration is the bias in the measurements, since a bias in the accelerometers and gyros turns into significant drift in the Euler Angles, position, and velocity over time. The effects of this bias are mitigated by taking measurements over a one minute period while not flying the quadrotor. A time average of those measurements is taken and subtracted from all future measurements before integrating to determine the Euler Angles, position, and velocity. This effectively removes the bias. While this greatly improves the drift of the integrated measurements, the integration of noisy measurements will still result in drift over time.

4.3.2 Kalman Filter.

A Kalman Filter is essentially an estimation routine which takes into account measurement noise characteristics, a linear model, and noise characteristics of the

model in order to estimate states of interest, or in the case of the quadrotor, the angular rates. The estimated angular rates can then be integrated through discrete DCM propagation according to Equations 8 to determine the quadrotor's attitude according to 41. This section is a summary of the work by Peter S. Maybeck on Kalman Filtering [16]. Some assumptions the Kalman Filter makes is that the linear model accurately describes the system and that all of the noise sources are zero mean, white, Gaussian, and uncorrelated. The Kalman Filter is proven to be optimal in a least squares sense based on these assumptions. The Kalman Filtering equations are presented here without proof or derivation. The discrete time system propagation and measurement equations are:

$$\begin{aligned} \vec{x}(k) &= \Phi\vec{x}(k-1) + B_d\vec{u}(k-1) + \vec{w}(k-1) \\ \vec{z}(k) &= H\vec{x}(k) + \vec{v} \end{aligned} \tag{47}$$

The state transition matrix Φ, discrete time input matrix B_d, and measurement matrix H are derived by discretizing the linear model. The state vector was previously defined in Equation 45 and the input vector was defined in Equation 46. The process noise vector $\vec{w}$ is zero mean, white Gaussian noise with covariance:

$$cov(\vec{w}) = Q \tag{48}$$

The measurement noise vector $\vec{v}$ is zero mean, white Gaussian noise with covariance:

$$cov(\vec{v}) = R \tag{49}$$

Since the measured values are assumed to be direct measurements of the state vector with additive noise, the measurement matrix is simply the identity matrix. The $\vec{z}$ is the measurement vector. The process and measurement noise are assumed to be

uncorrelated.

The first Kalman Filtering step is the propagation step:

$$\begin{aligned} \hat{\vec{x}}^{-}(k) &= \Phi(k-1)\hat{\vec{x}}^{+}(k-1) + B_d(k-1)\vec{u}(k-1) \\ P^{-}(k) &= \Phi(k-1)P^{+}(k-1)\Phi^{T}(k-1) + Q(k-1) \end{aligned} \tag{50}$$

It generates an estimate of what the state is expected to be at the next time step based on the linear model. It also generates an uncertainty in that estimate represented as a covariance matrix P. The vector $\hat{\vec{x}}^{-}$ is the state estimate and P^{-} is the covariance estimate before a measurement is taken. The vector $\hat{\vec{x}}^{+}$ is the state estimate and P^{+} is the covariance estimate after a measurement is taken.

The second Kalman Filtering step is the measurement update step:

$$\begin{aligned} K(k) &= P^{-}(k)H^{T}(k)\left[H(k)P^{-}(k)H^{T}(k) + R(k)\right]^{-1} \\ \hat{\vec{x}}^{+}(k) &= \hat{\vec{x}}^{-}(k) + K(k)\left(z(k) - H(k)\hat{\vec{x}}^{-}(k)\right) \\ P^{+}(k) &= \left(I - K(k)H(k)\right)P^{-1}(k) \end{aligned} \tag{51}$$

It generates an estimate of what the state is after a measurement based on the state estimate before the measurement and the measurement. The variable K is referred to as the Kalman Gain. The I is the identity matrix. The $z(k) - H(k)\hat{\vec{x}}^{-}(k)$ term is commonly called the residual and is the difference between the actual measurement and the predicted measurement based upon the results of the propagation step.

So long as the linear model supplied accurately describes the non-linear model, the Kalman Filter will provide better state estimates than the measurements alone. However, if the linear model is poor, it is better to use the measurements alone and not have the overhead of running a Kalman Filter. For the quadrotor, the measurement noise covariance matrix is well known since a noise characterization was generated in Section 4.3.1. Unfortunately, at this point in time, the process noise

is not well known, so instead, the process noise matrix is "tuned" to achieve smaller residuals since residuals are indicative of the errors the Kalman Filter is making in its estimation. The process noise matrix is also "tuned" to minimize the difference between the Kalman Filter estimates and the non-linear model.

Another important note is that since the linear models supplied are not the true systems, but linearized versions of the true non-linear systems with delta inputs delta states, the Kalman Filters must be Linearized Kalman Filters to match. This does not affect the Kalman Filtering equations already discussed. The only change is that the inputs to and outputs from the Linearized Kalman Filters are delta inputs and delta states.

V. Results and Analysis

5.1 Measurement Noise Characterization Results

To begin, the Allan Variance Analysis method of characterizing the measurement noise results are shown. Figures 13 and 14 show the results of the Allan Variance Analysis on the 48 hour inertial measurement unit (IMU) data collect. Figure 13 shows the gyro results and Figure 14 shows the accelerometer results. Figures 15 and 16 show the results of estimating the random walk and bias instability and plotting it on the same chart as the original Allan Variance results. While the general shape is preserved, the estimation does not fully capture the Allan Variance of the data. The next technique used to characterize the noise was power spectral density (PSD) matching. Figures 17, 18, 19, 20, 21 and , 22 show the PSD matching results for the x-, y-, and z-axis gyros and accelerometers respectively. The transfer functions calculated from examination of the PSD plots and used to simulate the noise shown in green on the PSD plots are:

$$\begin{aligned} H_{arx}(s) &= \tfrac{0.9(s+0.015)}{s^2+150s} \\ H_{ary}(s) &= \tfrac{0.9(s+0.015)}{s^2+150s} \\ H_{arz}(s) &= \tfrac{0.8(s+0.0175)}{s^2+140s} \\ H_{ax}(s) &= \tfrac{0.1(s+0.125)}{s^2+110s} \\ H_{ay}(s) &= \tfrac{0.1(s+0.275)}{s^2+125s} \\ H_{az}(s) &= \tfrac{0.1(s+10)}{s^2+120s} \end{aligned} \tag{52}$$

where the variables H_{arx}, H_{ary}, H_{arz}, H_{ax}, H_{ay}, and H_{az} are the x-, y-, and z-axis gyro and accelerometer transfer functions respectively.

In order to compare these results to the Allan Variance results, an Allan Variance Analysis was performed on the simulated data and plotted against the Allan Variance

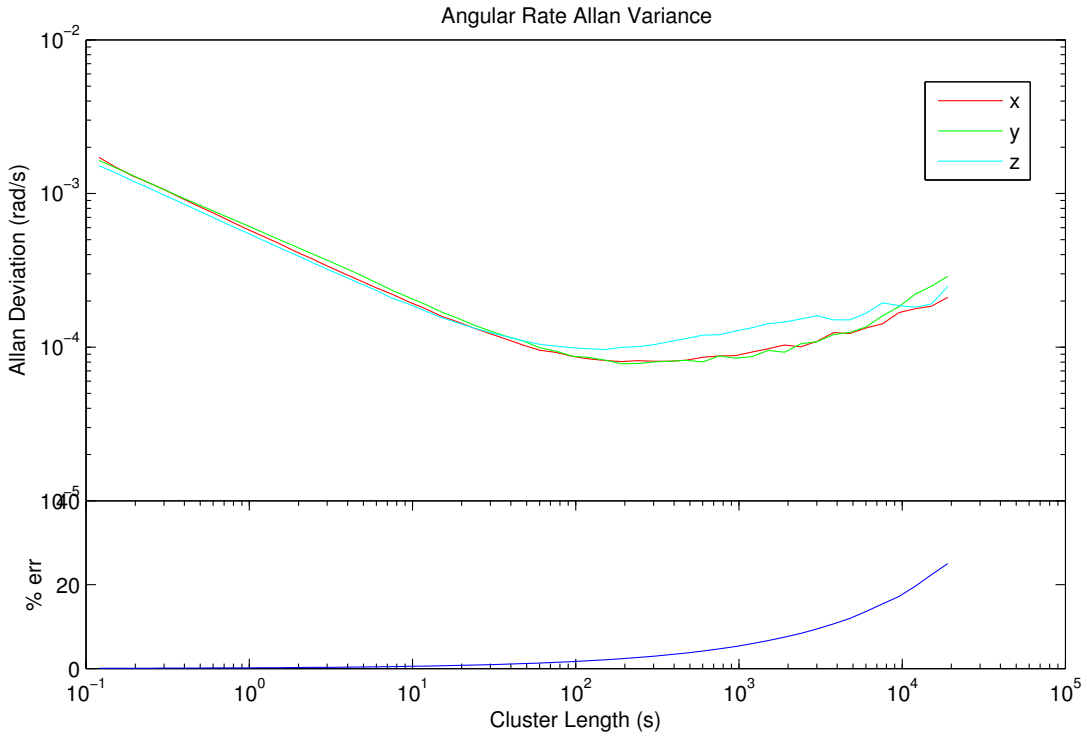

Figure 13. 48 Hour IMU Data Collect, Gyro Allan Variance Analysis Results.

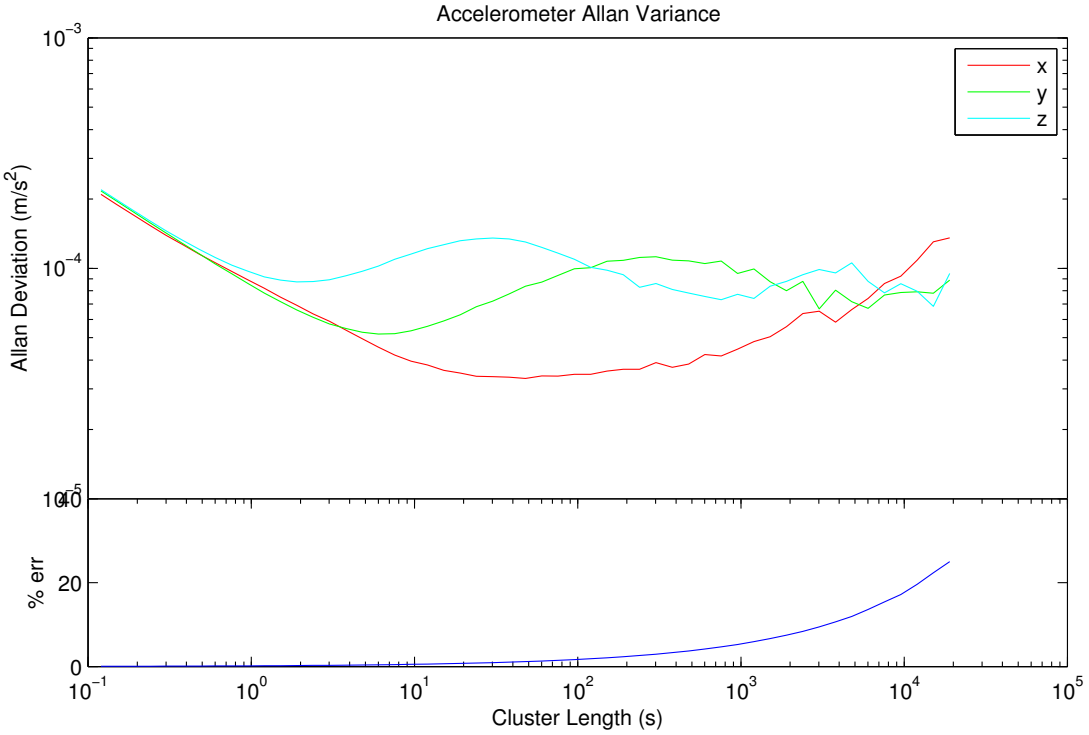

Figure 14. 48 Hour IMU Data Collect, Accelerometer Allan Variance Analysis Results.

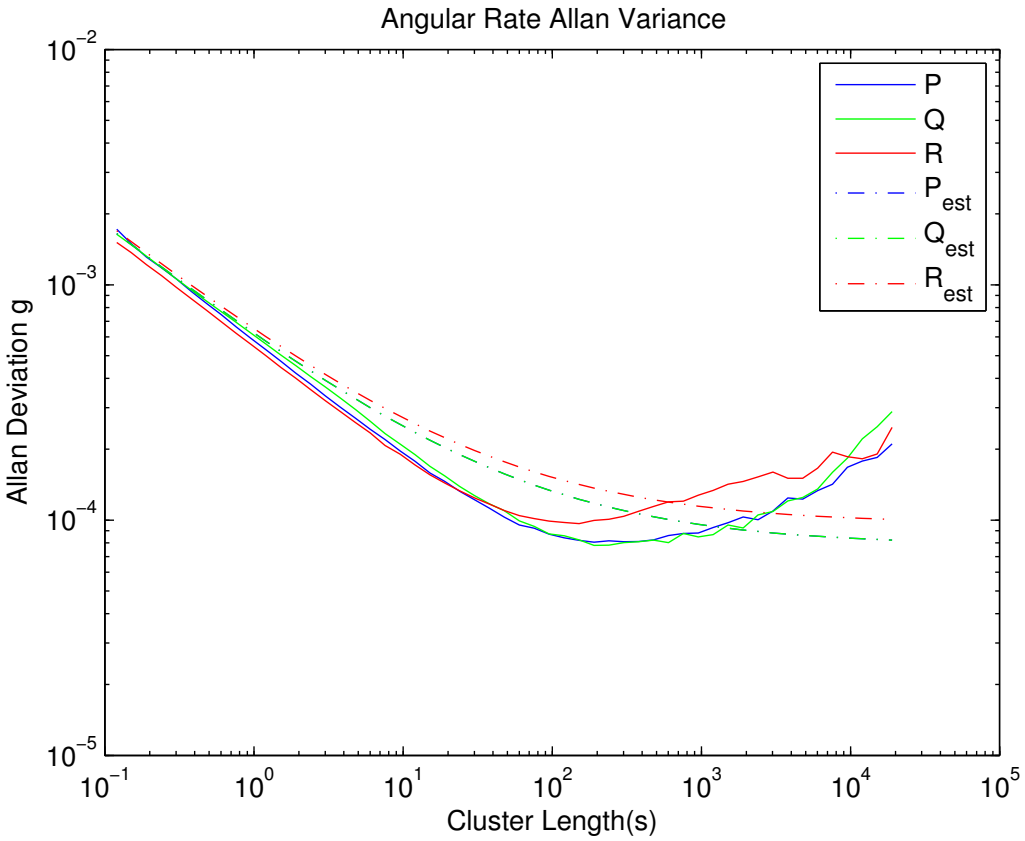

Figure 15. Gyro Allan Variance Bias Instability Estimation.

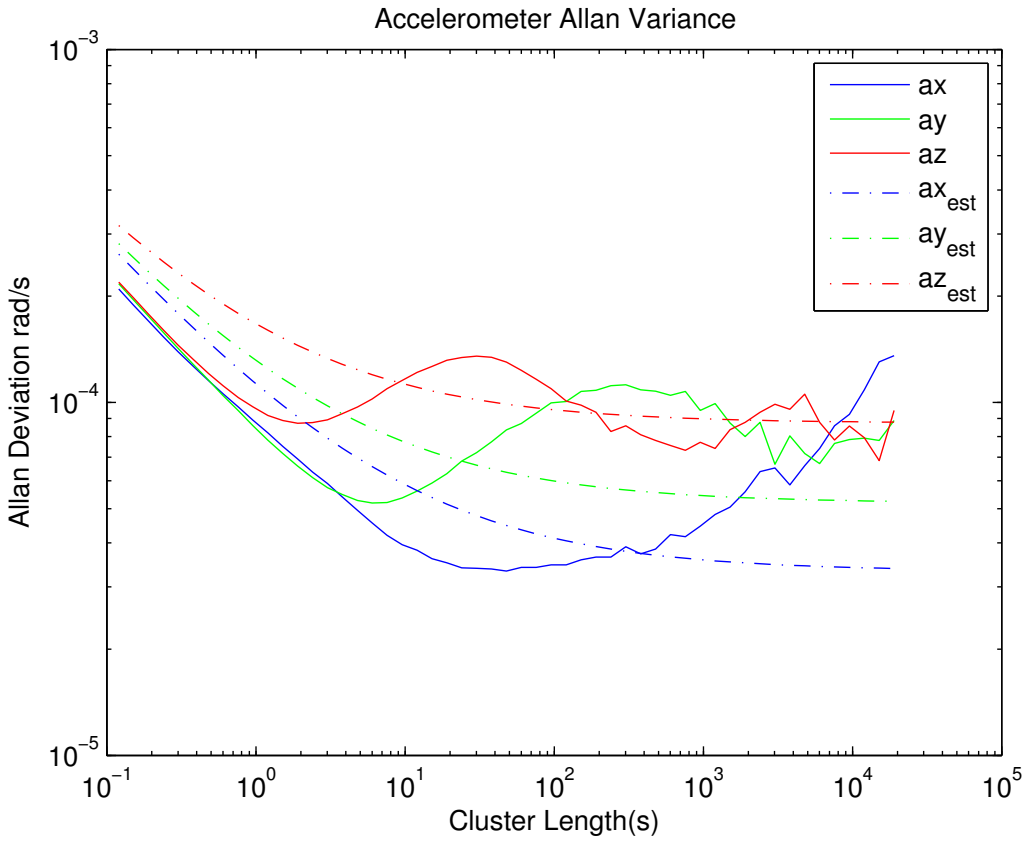

Figure 16. Accelerometer Allan Variance Bias Instability Estimation.

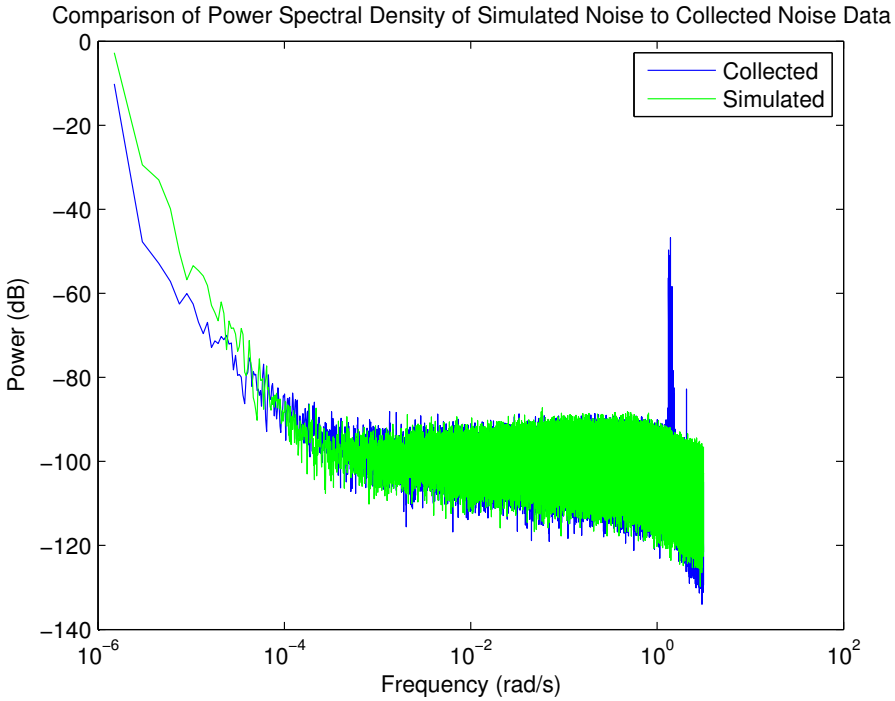

Figure 17. Power Spectral Density of Collected and Simulated Data for x-axis Gyro.

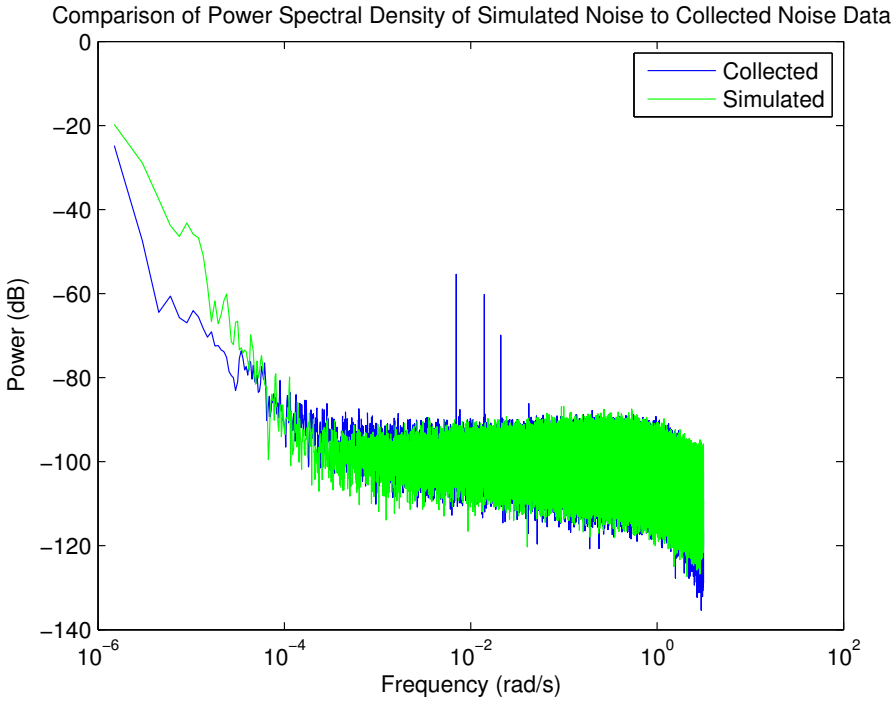

Figure 18. Power Spectral Density of Collected and Simulated Data for y-axis Gyro.

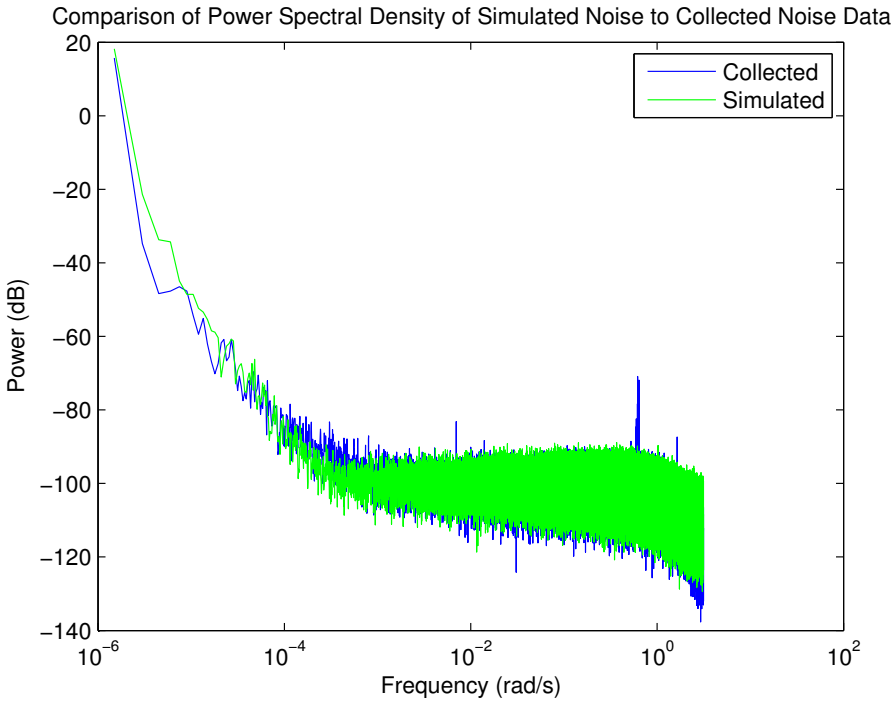

Figure 19. Power Spectral Density of Collected and Simulated Data for z-axis Gyro.

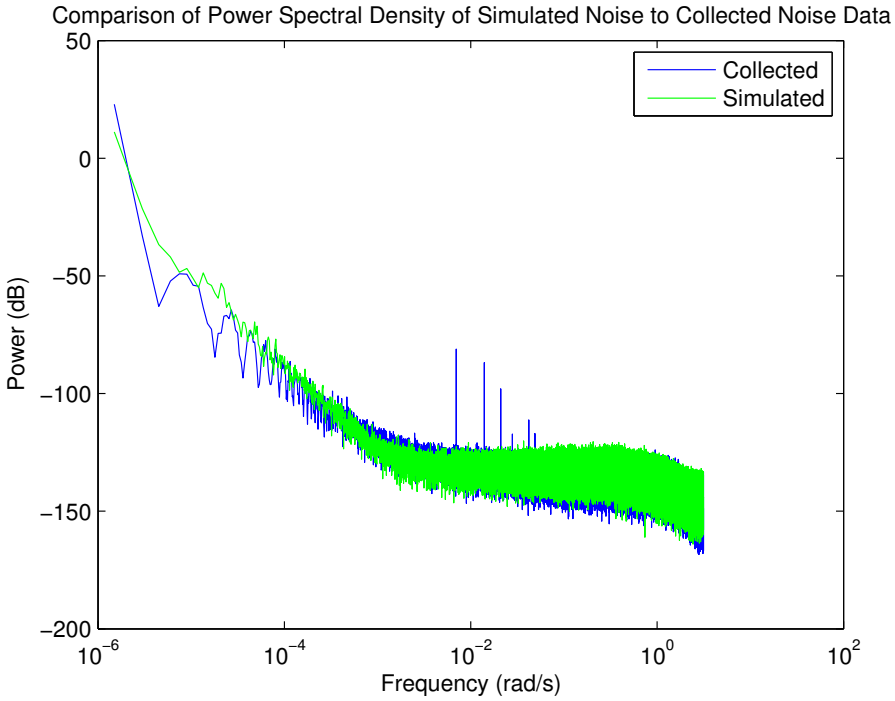

Figure 20. Power Spectral Density of Collected and Simulated Data for x-axis Accelerometer.

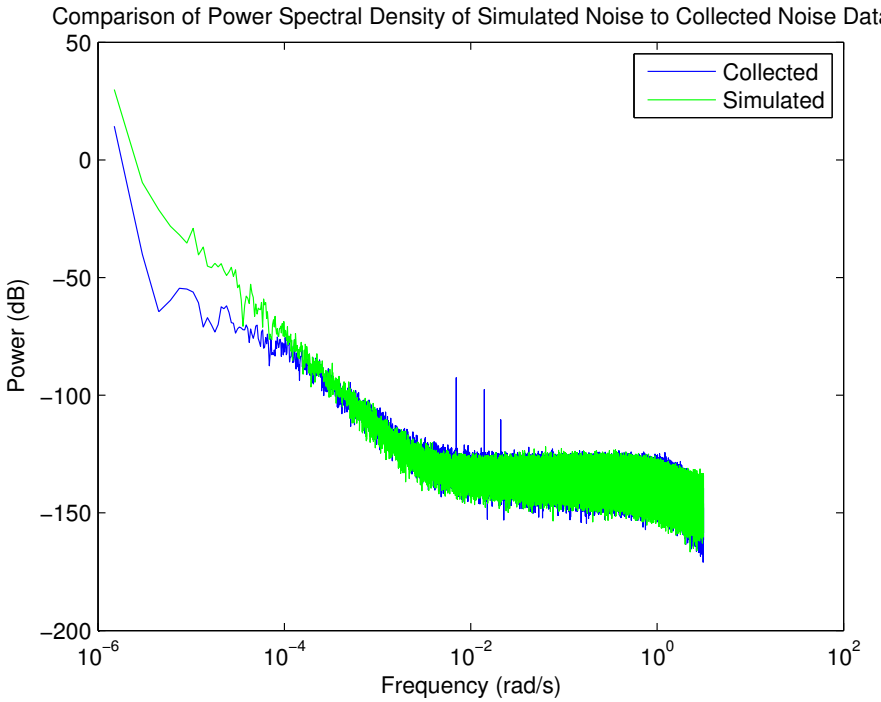

Figure 21. Power Spectral Density of Collected and Simulated Data for y-axis Accelerometer.

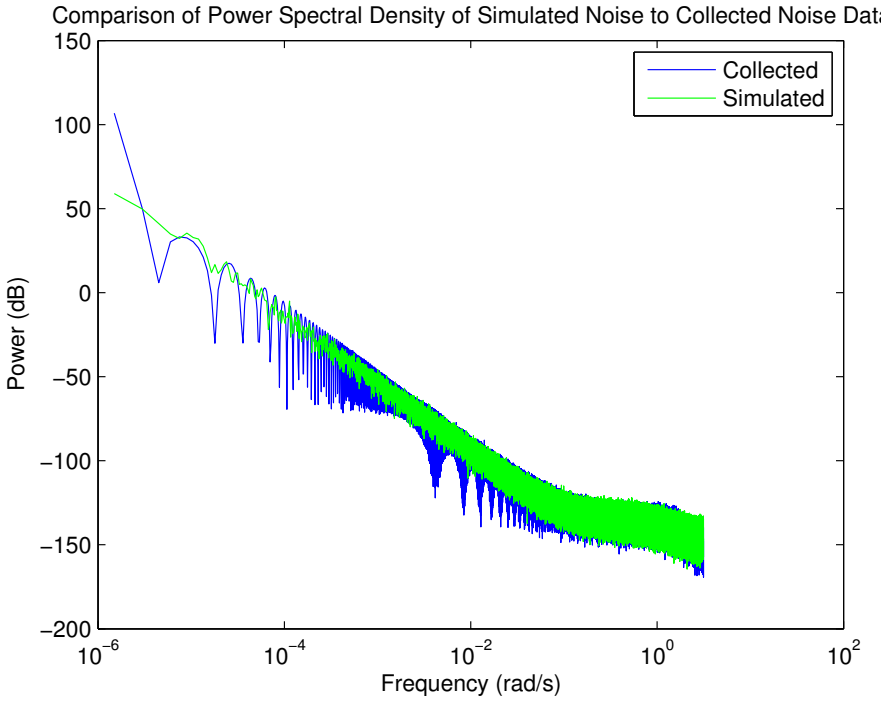

Figure 22. Power Spectral Density of Collected and Simulated Data for z-axis Accelerometer.

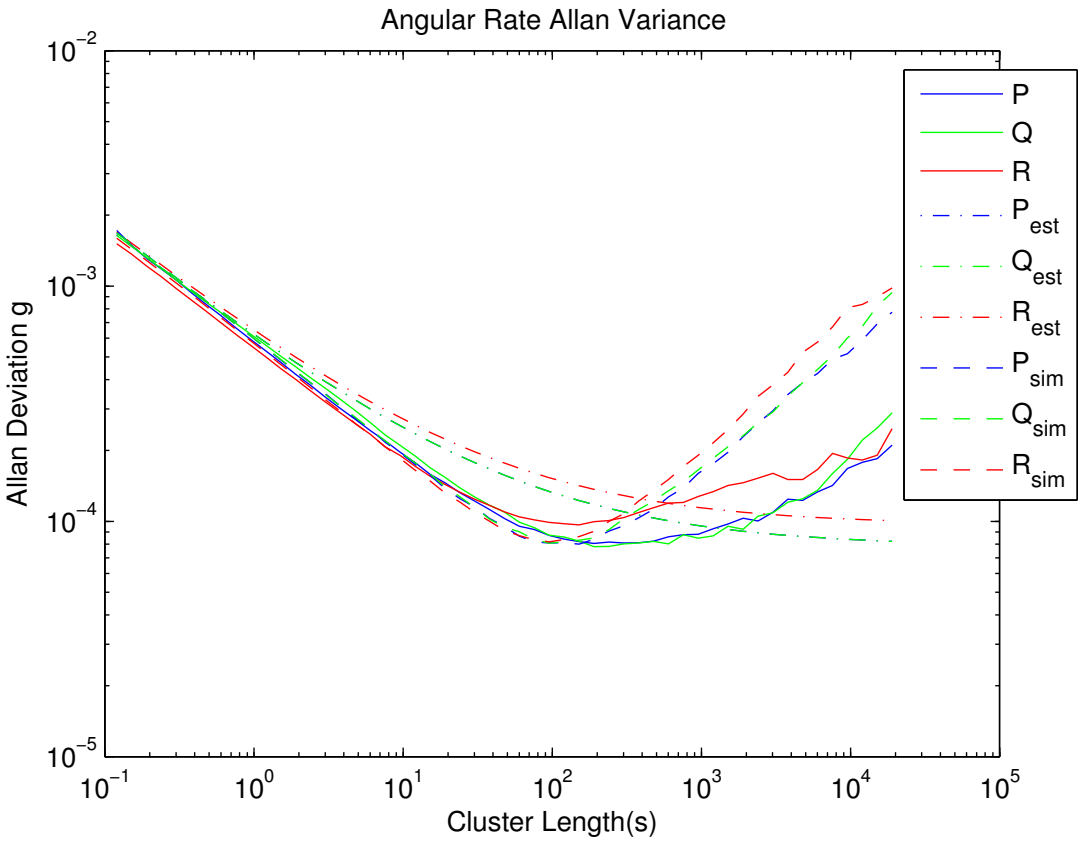

Figure 23. Gyro Allan Variance Bias Instability Estimation and Simulated Data.

Analysis of the collected data and the random walk and bias instability estimate. These plots are shown in Figures 23 and 24. These last two Allan Variance plots show that the PSD analysis method captures a little more than the Bias Instability estimation, but still does not accurately represent the real IMU noise, particularly for the z^b-axis accelerometer. This is why real IMU noise is used in the simulations rather than using simulated noise. The 48 hours of data was broken into 30 minute segments and is used in the simulations.

5.2 Non-linear Model Simulation Results

For each of the plots shown in this section, a total of 30 Monte-Carlo simulations were run using real noise collected from the IMU to corrupt the measurements taken

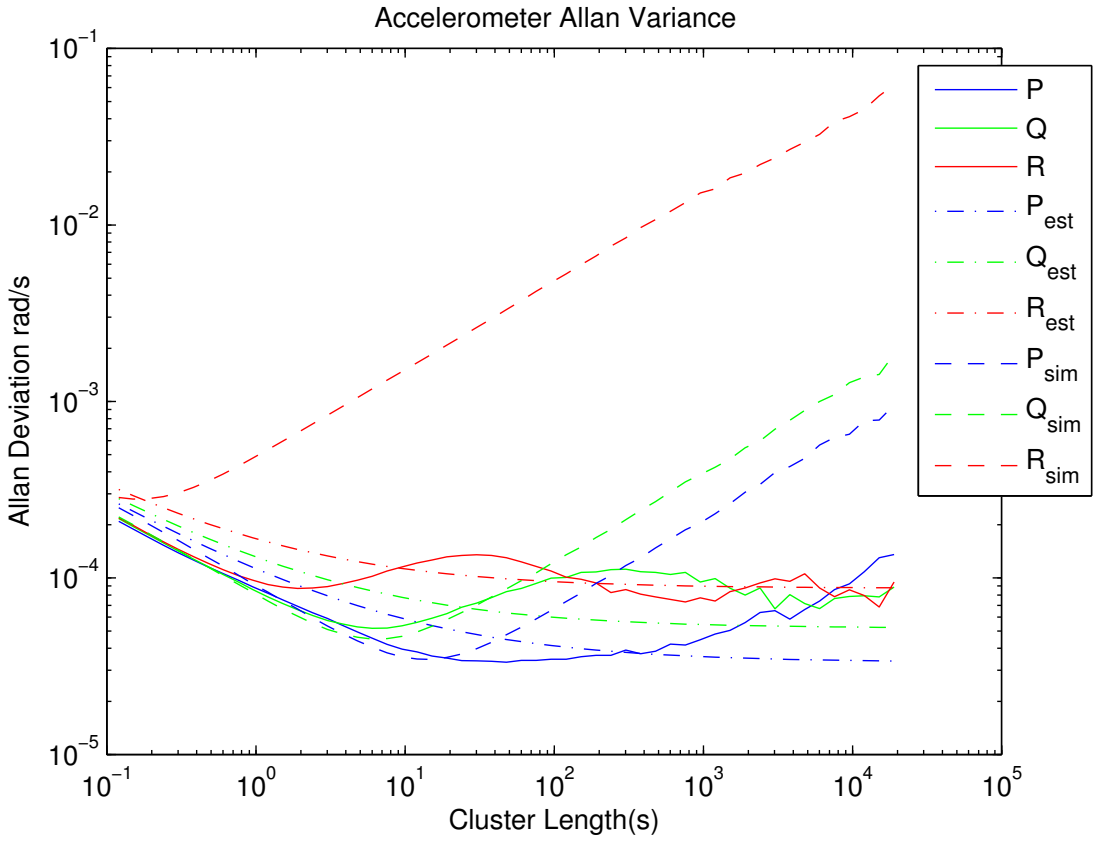

Figure 24. Accelerometer Allan Variance Bias Instability Estimation and Simulated Data.

from the non-linear truth model as mentioned previously in Section 4.3.1. Each of the 30 Monte-Carlo simulations used a different 30 minute noise period from the 48 hours of data.

Figure 25 shows the static simulation results for the simple non-linear model when a hover is commanded for 5 minutes after an initial one minute bias removal period where the bias is removed from the measurements. As mentioned in Section 4.1, the inputs to the control loop are the measured values which is why it appears that the true values drift while the measured values remain near the reference command. This simply means that the mean and standard deviation of the true values with respect to the measured values is indicative of the measurement drift. The "true" values are the non-linear model values.

Figure 26 shows the dynamic simulation results for the non-linear model when small commands are issued on the roll and pitch channels after a one minute bias removal period where the bias is removed from the measurements. While it is not shown in this figure, a doublet throttle command is issued in order to simulate take off from the ground shortly before the roll and pitch commands are issued. Some of the cross coupling in the equations of motion result in some noise appearing shortly before the first roll command is issued, particularly on the yaw channel. In both of these simulations, the mixer matrix from Equation 15 is used and there is no laser detection and ranging (LADAR) payload.

However, this model fails to capture what would happen if the quadrotor were not perfectly balanced or the motors were not placed equidistant from the center of gravity. The computation of the center of gravity in Equation 22 shows that there is an offset in the center of gravity from where the assumed center of gravity was. Therefore, a more complex non-linear model which accounts for that offset is proposed. The more complex non-linear model differs from the simpler non-linear

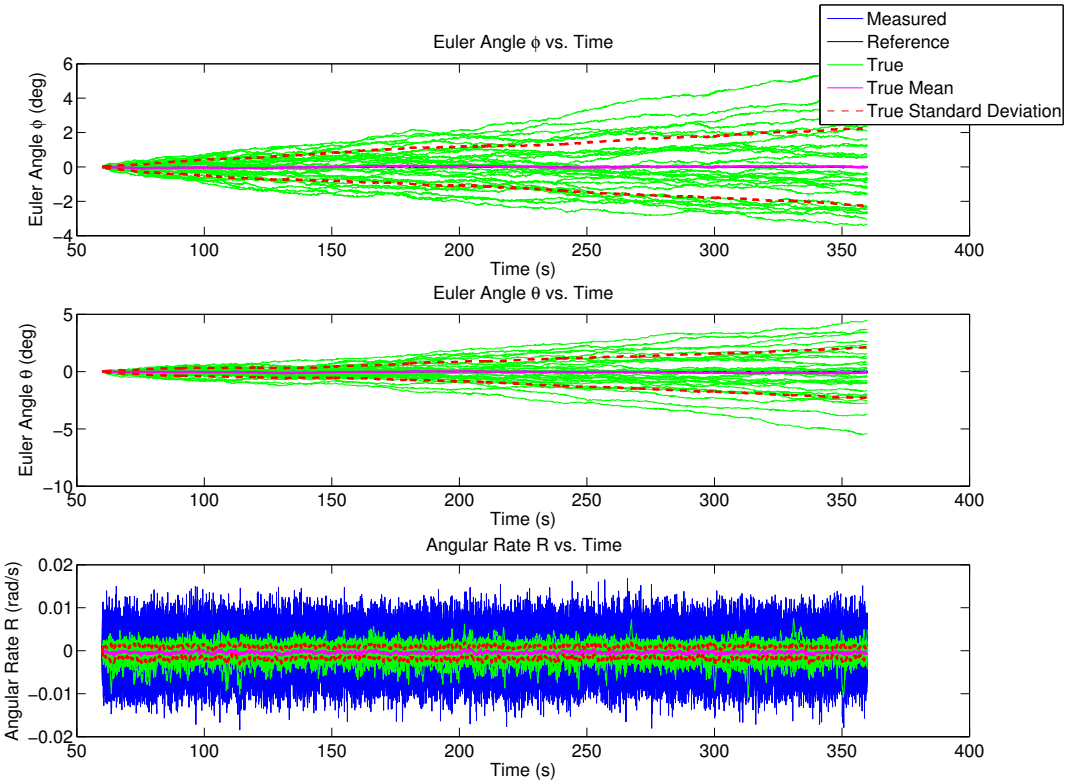

Figure 25. Simple Non-Linear Model Static Simulation Results with Standard Mixer Matrix.

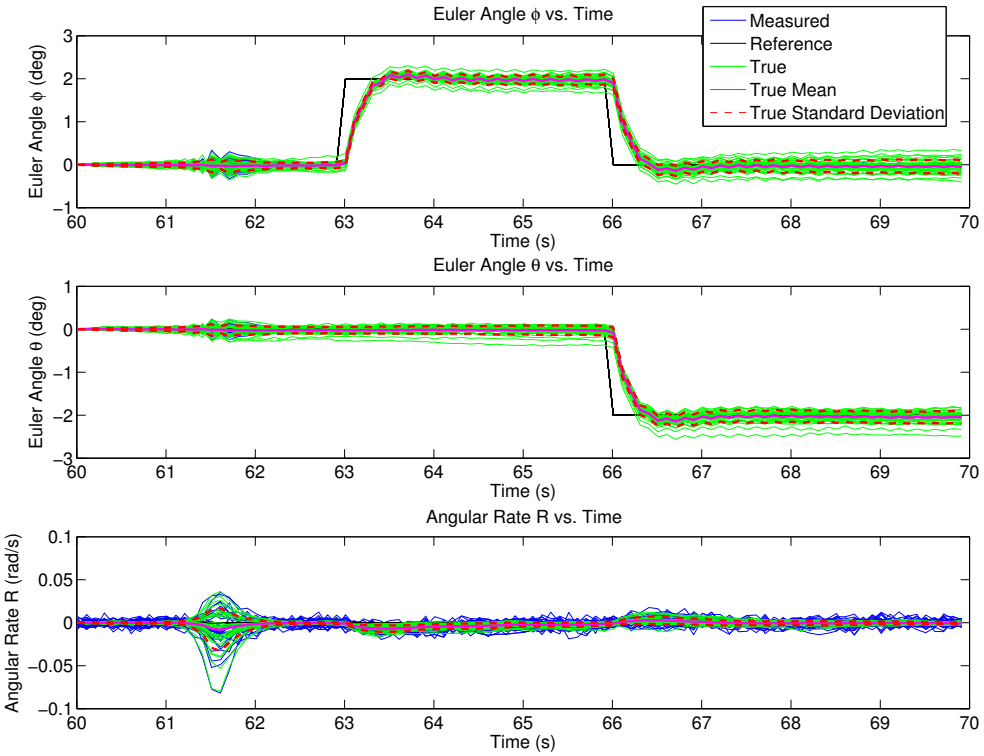

Figure 26. Simple Non-Linear Model Dynamic Simulation Results with Standard Mixer Matrix.

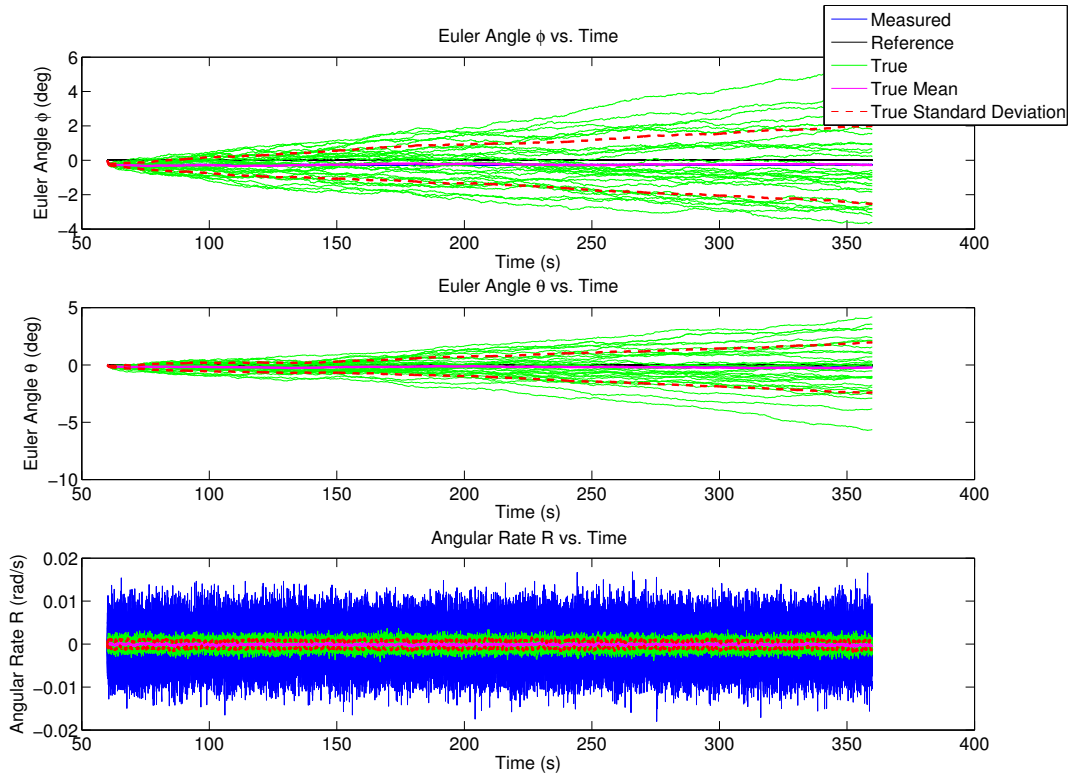

Figure 27. Complex Non-Linear Model Static Simulation Results with Standard Mixer Matrix.

model in Equation 33 as shown:

$$\begin{aligned} \sum L &= F_1(-y_p(1)) + F_2(-y_p(2)) + F_3(-y_p(3)) + F_4(-y_p(4)) \\ \sum M &= F_1 x_p(1) + F_2 x_p(2) + F_3 x_p(3) + F_4 x_p(4) \\ \sum N &= \tau_1 - \tau_2 + \tau_3 - \tau_4 \end{aligned} \tag{53}$$

The other equations describing the non-linear model remain the same. Note that in this new model, $x_p(i)$ and $y_p(i)$ refer to the x and y position of the i^{th} motor relative to the computed center of gravity.

Figures 27 and 28 show the same simulation results for the more complex non-linear model using the standard mixer matrix. The primary difference in the simple and complex non-linear models is a steady state error that rapidly develops in the

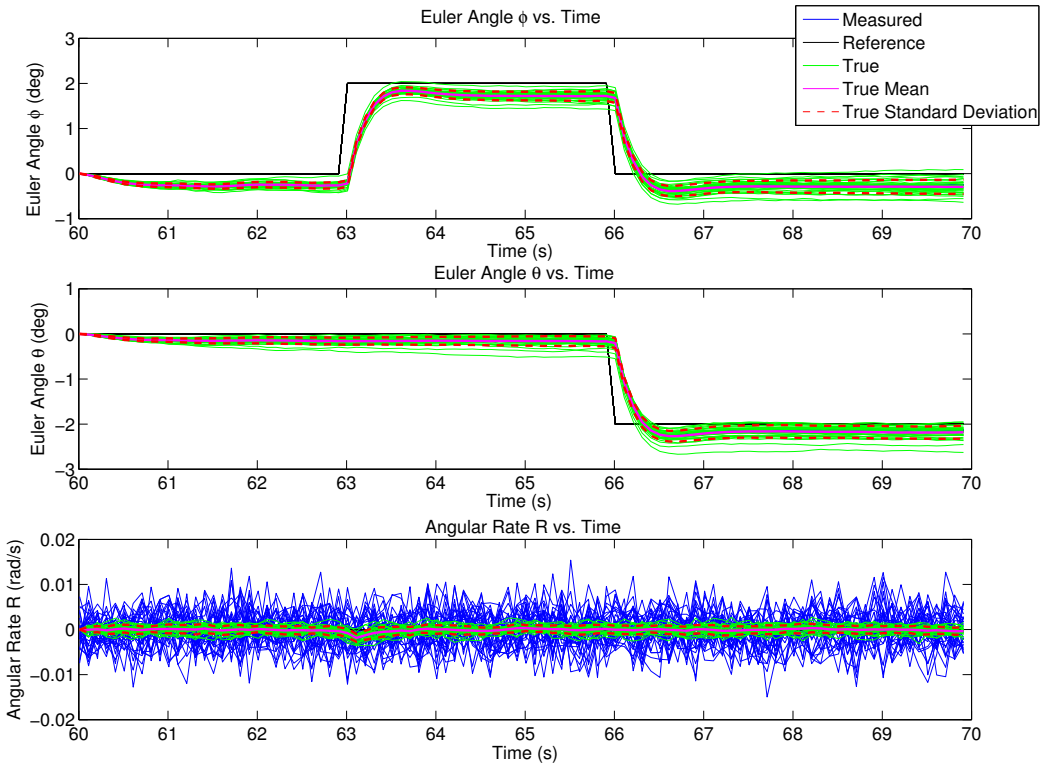

Figure 28. Complex Non-Linear Model Dynamic Simulation Results with Standard Mixer Matrix.

more complex non-linear model. This can be explained by the fact that the commands sent to the motors based on the mixer matrix in 15 assume that the quadrotor is perfectly balanced and each motor is equidistant from the center of the quadrotor. As a result, the quadrotor experiences some initial roll and pitch for which the control loop is compensating. Essentially, the initial hover command is seen as a step input by the system with resulting steady state error. This complex model paired with this mixer matrix is assumed to be the truth model.

5.3 Linear Model Validation Against Non-Linear Model

This section is very similar to Section 5.2 in terms of the figures that will be examined. The exact same static and dynamic flight profiles will be used to compare the linear model results to the non-linear model results. Each figure will also show 30 Monte-Carlo runs for the 30 different sets of real IMU noise.

Figures 29 and 30 show a comparison between the linear model from Section 4.2.1 and the non-linear model from Section 4.1. The only difference between these two figures and Figures 25 and 26 is the addition of the linear model data. While the mean of the linear model is very close to the true mean for the roll and pitch channels, the variance is terrible. This is because linear models do not typically reject noise well.

Looking at the static profile shows that the linear model exhibits some interesting sinusoidal tendencies in the roll and pitch channels. Another point of interest is how all of the Monte-Carlo runs show the linear model yaw channel drifting in the positive direction. While the exceptionally poor performance of the yaw channel alone invalidates this linear model, the standard deviation drift of the roll and pitch channels, particularly when looking at the dynamic flight profile, also invalidates this linear model according to the definition in Section 4.2.2. If it were possible to ignore

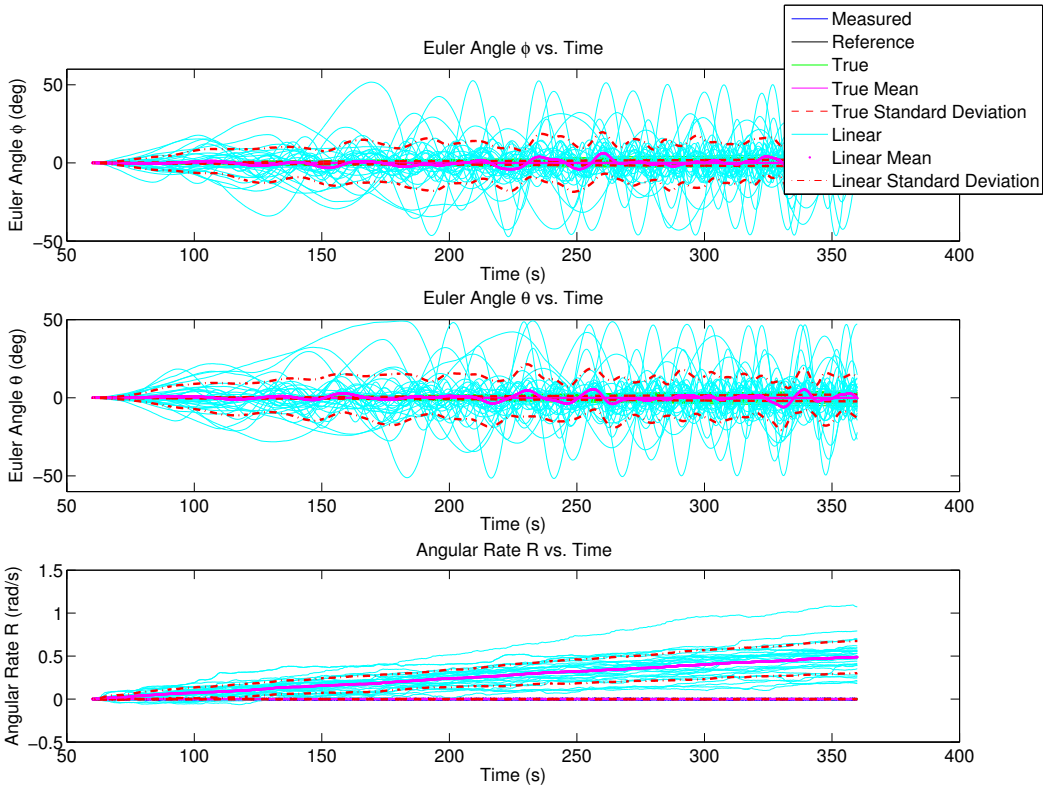

Figure 29. Linear Model Static Simulation Results with Standard Mixer Matrix.

the yaw channel and the standard deviation drift in the roll and pitch channels and only to look at the mean, this would appear to be a valid model. In fact, if this linear model is simulated in the absence of noise, it very closely follows the non-linear model. Figures 31 and 32 demonstrate this.

However, this linear model does not reflect how a linear model based on the more complex non-linear model will perform. A second linear model is developed based on Equations 25, 26, 28, 2 and 53 which were used to derive the complex non-linear model. Figures 33 and 34 show a comparison between the complex non-linear model and second linear model where the standard mixer matrix is used and noise has been turned back on. It is immediately obvious that this linear model is very poor as it begins oscillating almost immediately and quickly diverges without any

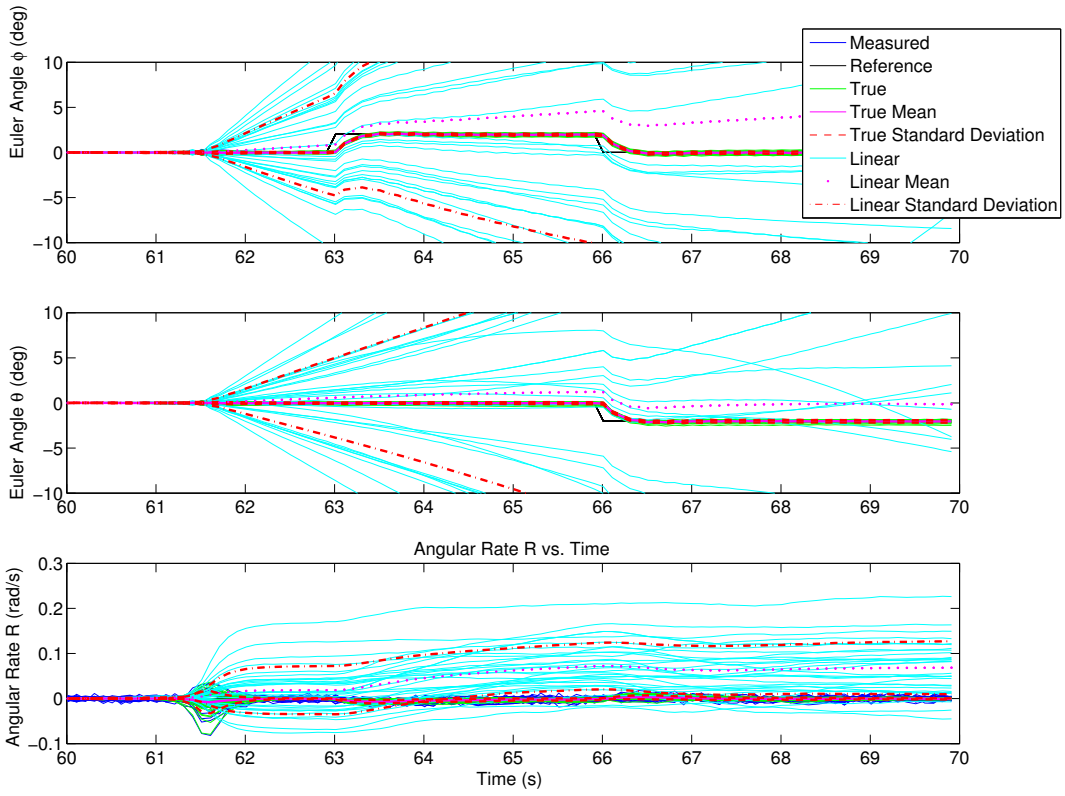

Figure 30. Linear Model Dynamic Simulation Results with Standard Mixer Matrix.

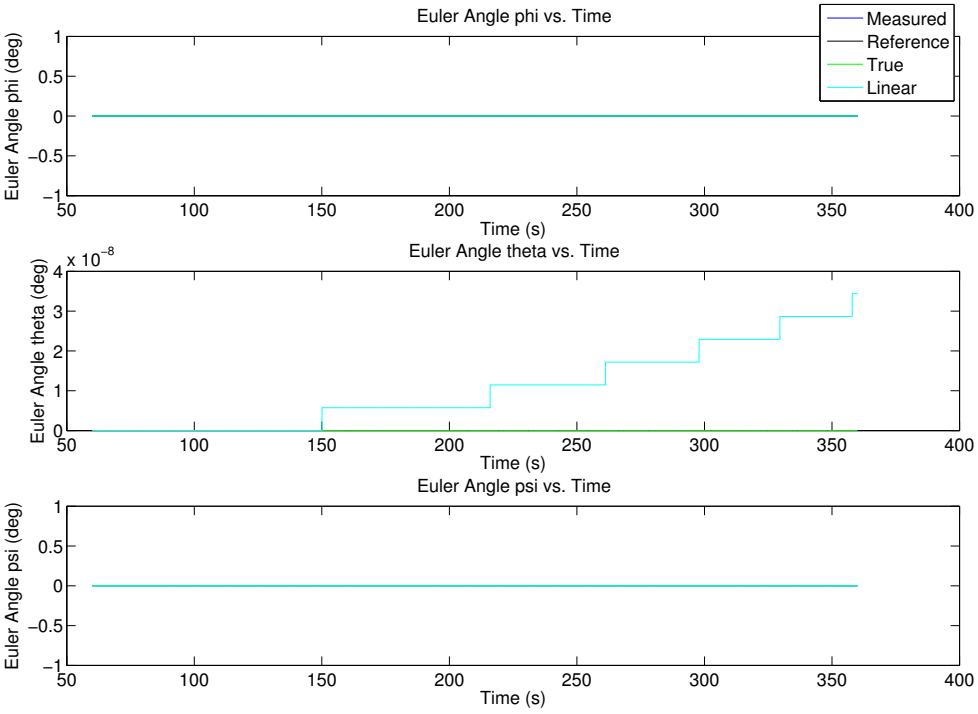

Figure 31. Linear Model Static Simulation Results with Standard Mixer Matrix and No Noise.

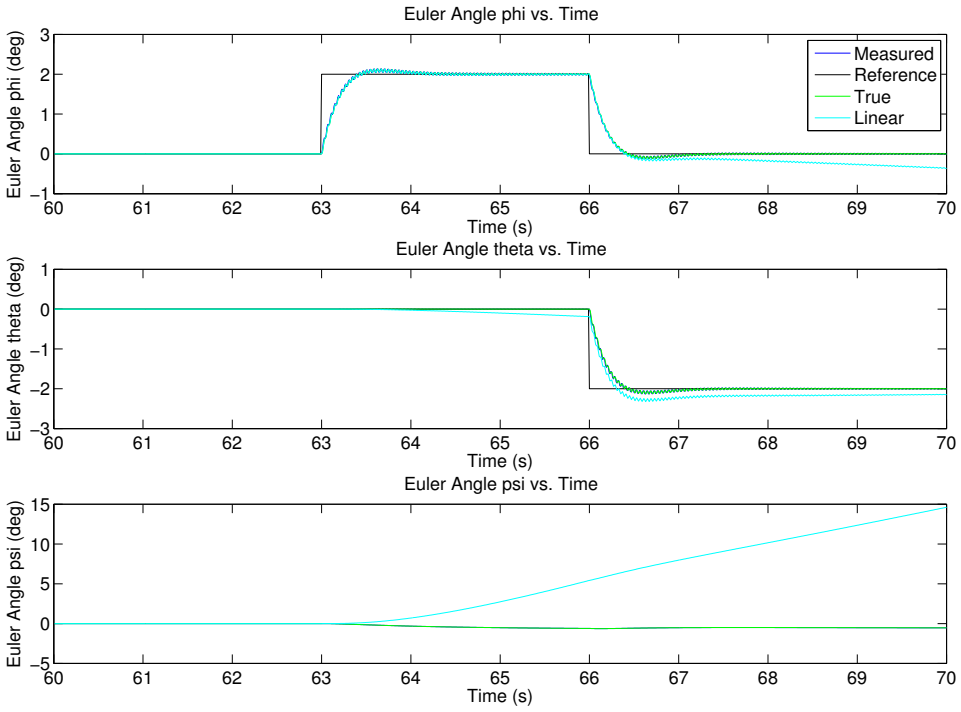

Figure 32. Linear Model Dynamic Simulation Results with Standard Mixer Matrix and No Noise.

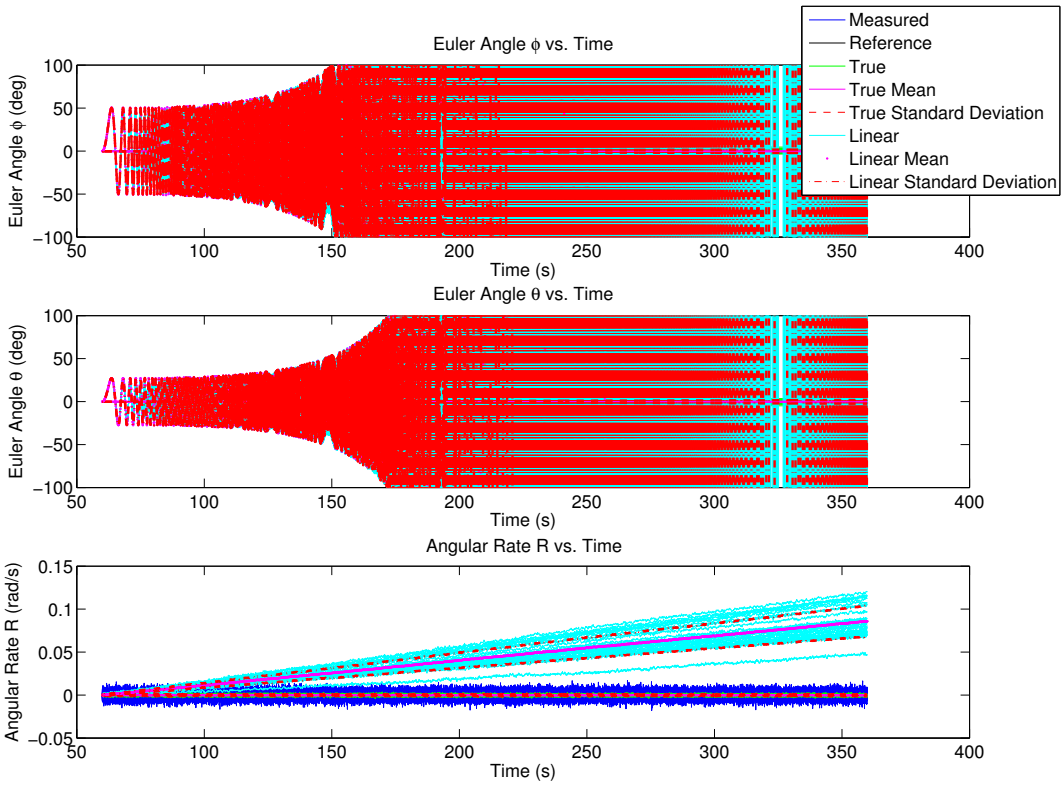

Figure 33. Linear Model 2 Static Simulation Results with Standard Mixer Matrix.

apparent bound. The most likely cause for this is that this linear model is based on the assumption that each rotor should supply one-fourth the thrust needed to maintain a hover and that the quadrotor is perfectly balanced. The nominal motor commands do not account for the fact that the motors are not perfectly positioned. Since the motors are not perfectly positioned, these nominal commands will result in the quadrotor rolling and/or pitching.

A third linear model is developed in an attempt to correct the problems with the second linear model. The target of this model is determining accurate nominal input values to the linear model. It was determined that the best way to determine these nominal values is to compute a new mixer matrix in an intelligent fashion, multiply the inverse mixer matrix by the desired roll, pitch, yaw, thrust vector for a hover,

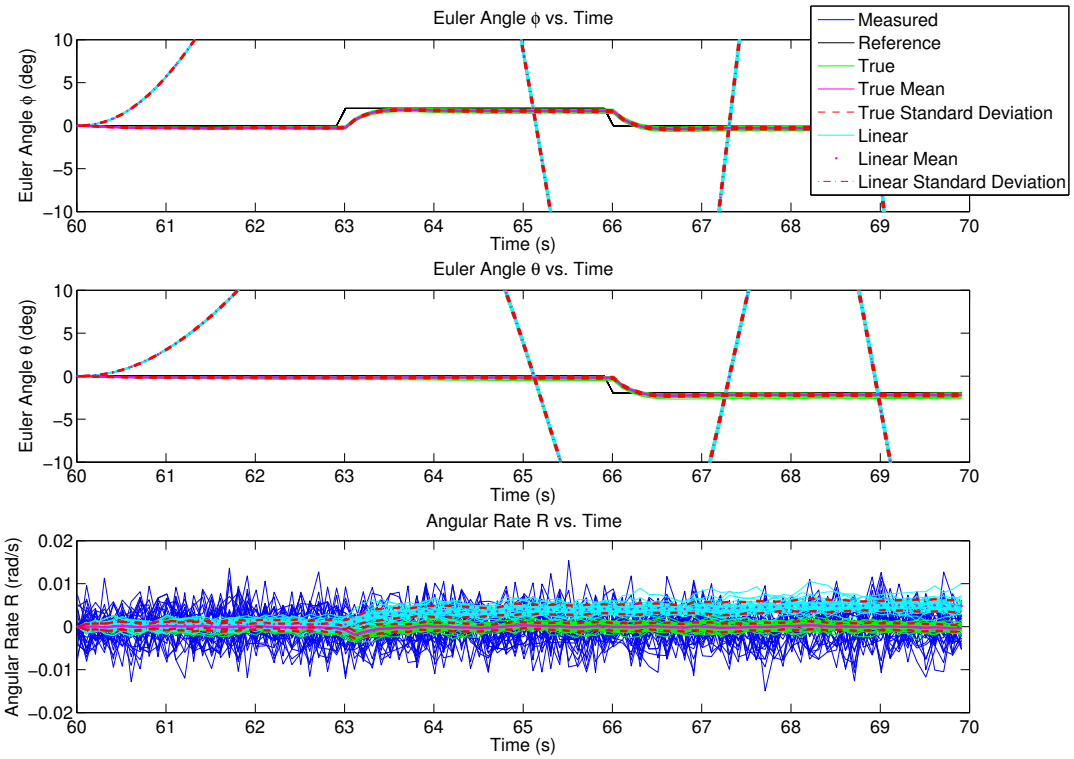

Figure 34. Linear Model 2 Dynamic Simulation Results with Standard Mixer Matrix.

and extract the desired nominal inputs.

To understand how to determine a new mixer matrix, it is useful to look at how the original mixer matrix was derived in Section 2.3.3 once again. After looking at this derivation, it is evident that the derivation of a new mixer matrix should have its derivation from Equations 10 and 53 since the original mixer matrix was based on Equations 10 and 11. While it appears that the yaw and thrust components of the mixer matrix should remain unchanged, it appears necessary to adjust the roll and pitch components. At the same time, in order to preserve the overall magnitude of values in the mixer matrix, it would be beneficial if the maximum and minimum values in the mixer matrix were $+1$ and -1. Looking back at Equation 11, it seems that the normalization factor is the distance of the motors from the center of the vehicle. For this more complex model, the normalization factor should be the maximum motor distance from the center of gravity for each axis. The normalization factors are:

$$\begin{aligned} n_x &= max(|x_p(1)|, |x_p(2)|, |x_p(3)|, |x_p(4)|) \\ n_y &= max(|y_p(1)|, |y_p(2)|, |y_p(3)|, |y_p(4)|) \end{aligned} \tag{54}$$

where n_x is the x-axis normalization factor and n_y is the y-axis normalization factor. The new mixer matrix is the computed as:

$$Mix = \begin{bmatrix} -\frac{y_p(1)}{n_y} & -\frac{y_p(2)}{n_y} & -\frac{y_p(3)}{n_y} & -\frac{y_p(4)}{n_y} \\ \frac{x_p(1)}{n_x} & \frac{x_p(2)}{n_x} & \frac{x_p(3)}{n_x} & \frac{x_p(4)}{n_x} \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{55}$$

The nominal inputs are then computed as:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = Mix^{-1} \begin{bmatrix} u_p \\ u_q \\ u_r \\ u_{thrust} \end{bmatrix} \tag{56}$$

which is identical to Equation 20. For simulation, this modified mixer matrix was used not only in computing nominal inputs for the linear model, it was also used in the non-linear simulation as well and replaced the original mixer matrix. Changing out the mixer matrix in the non-linear model has little impact since the control loops were already compensating for these errors.

Figures 35 and 36 show the results of this attempt when compared to the complex non-linear model. While it is not immediately evident, it can be seen by careful examination of the plots that this linear model diverges a little more slowly making it a little better, but not by much. It can also be seen that the initial direction of the first oscillation in the roll and pitch channels is opposite the direction of the first oscillation in the second linear model, which suggests that it is possible that this third linear model overcompensated for the errors in the second linear model.

A final linear model, which is very similar to the third linear model in that it is focused on the mixer matrix, is developed in an attempt to account for the errors in the second and third linear models, but is a complete improvisation. As such, it should not be implemented outside of simulation despite giving far better results than the second and third linear models. The final mixer matrix was computed under the assumption that the modified mixer matrix should be a blend between the original mixer matrix in Equation 15 and the modified mixer matrix in Equation 55 so that the mixer matrix would only account for a portion of the center of gravity offset

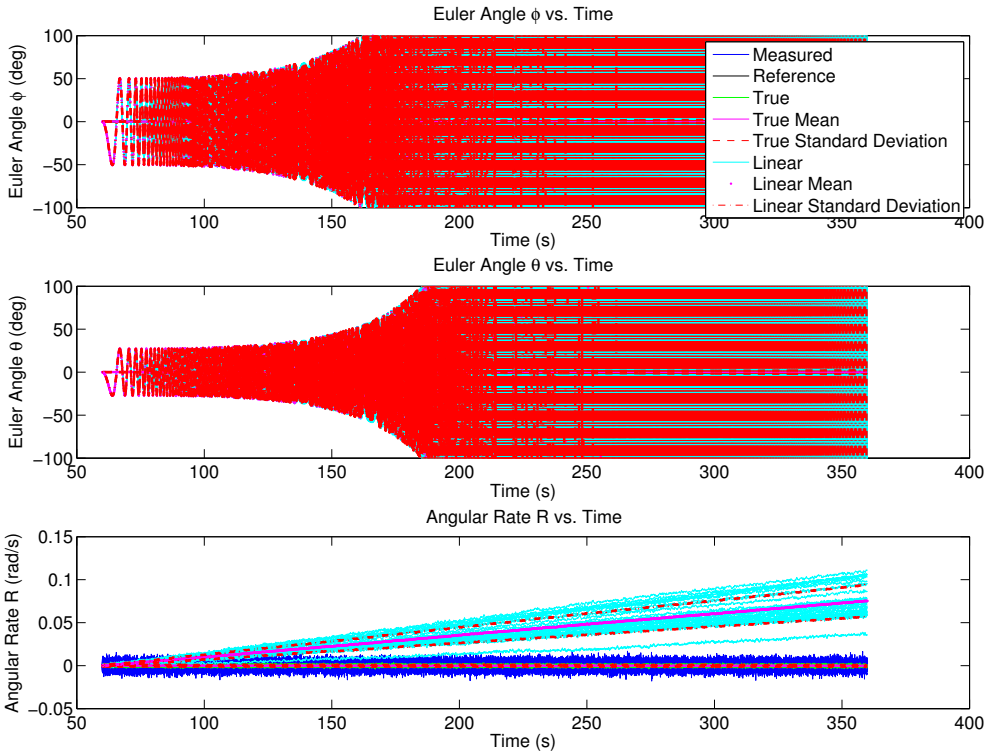

Figure 35. Linear Model 3 Static Simulation Results with First Modified Mixer Matrix.

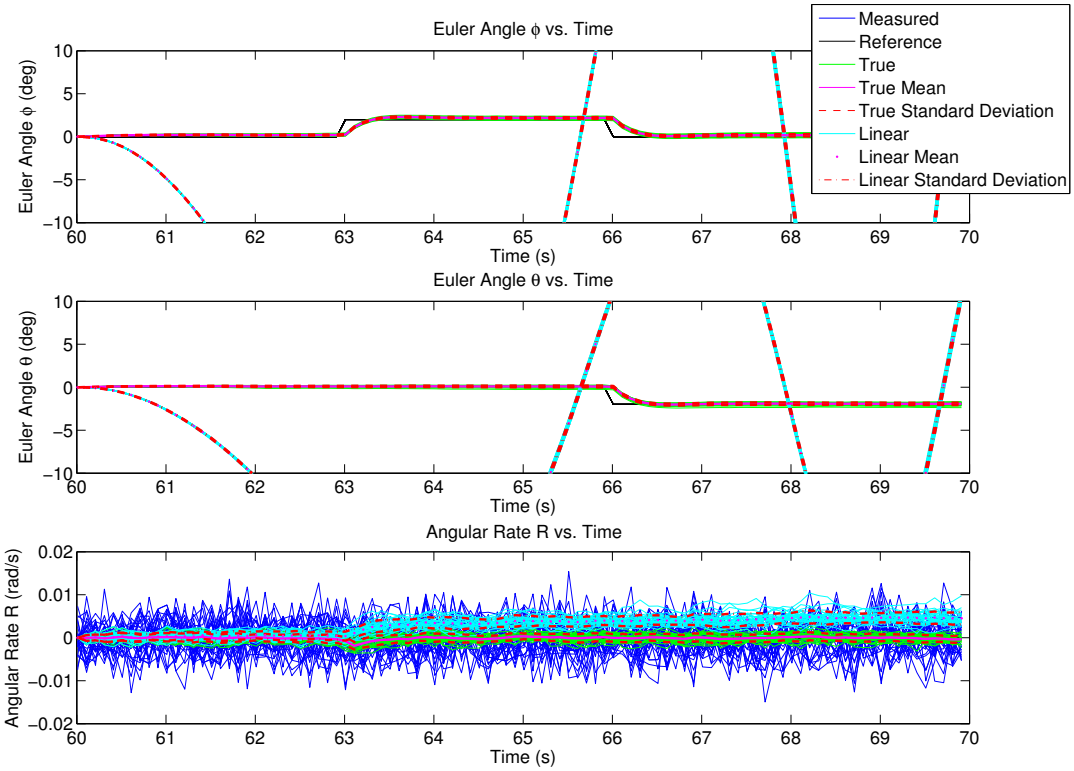

Figure 36. Linear Model 3 Dynamic Simulation Results with First Modified Mixer Matrix.

rather than not accounting for it or accounting for all of it. This mixer matrix was "tuned" until the linear model matched the non-linear model, with a small drift in the computed Euler Angles over time, similar in magnitude to the drift caused by integrating IMU measurements over time to determine Euler Angles. The final mixer matrix is:

$$Mix = \begin{bmatrix} -\frac{y_p(1)+\frac{CG_y}{2.183}}{n_y} & -\frac{y_p(2)+\frac{CG_y}{2.183}}{n_y} & -\frac{y_p(3)+\frac{CG_y}{2.183}}{n_y} & -\frac{y_p(4)+\frac{CG_y}{2.183}}{n_y} \\ \frac{x_p(1)+\frac{CG_x}{2.183}}{n_x} & \frac{x_p(2)+\frac{CG_x}{2.183}}{n_x} & \frac{x_p(3)+\frac{CG_x}{2.183}}{n_x} & \frac{x_p(4)+\frac{CG_x}{2.183}}{n_x} \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{57}$$

and was used in computing the nominal values for the linear model according to Equation 56 and also used in the non-linear model in simulation for comparison to this linear model. This linear model is highly dependent on the location of individual components. Adding the laser detection and rangin (LADAR) payload changes the scaling factor in Equation 57 from 2.183 to 2.24.

Figures 37 and 38 show the results of simulating this model. It is immediately evident that this linear model is far superior to all previous models with the exception of the first linear model, which it seems to be on par with. The main difference is that the oscillatory periods on this linear model appear longer than those on the first linear model; however, these oscillations more closely resemble those seen in the second and third linear models than in the first linear model.

Another interesting development is that it appears that the steady state error in the roll and pitch channels first observed in the second linear model has been greatly reduced, if not removed entirely. While this model does seem to offer greatly improved performance, it is not recommended for use on the physical quadrotor since it was generated using an ad-hoc approach and still appears to have the same instability

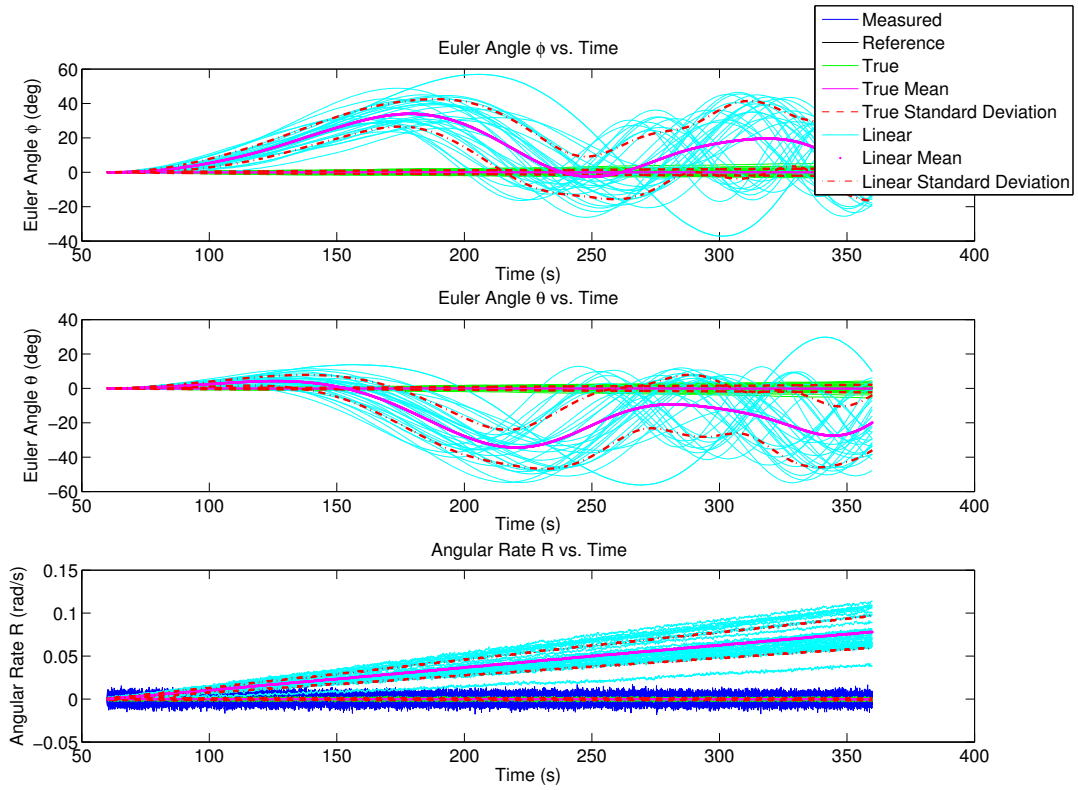

Figure 37. Linear Model 4 Static Simulation Results with Second Modified Mixer Matrix.

that was associated with the second and third linear models.

5.4 Kalman Filter Performance

Looking back at the results from Section 5.3, it is expected that the second and third linear models will make very poor models on which to base a Kalman Filter, therefore Kalman Filters were only implemented based on the first and fourth linear models. These Kalman Filters were implemented with a process noise matrix selected to be five times the magnitude of the measurement noise matrix as shown:

$$Q = 5R \tag{58}$$

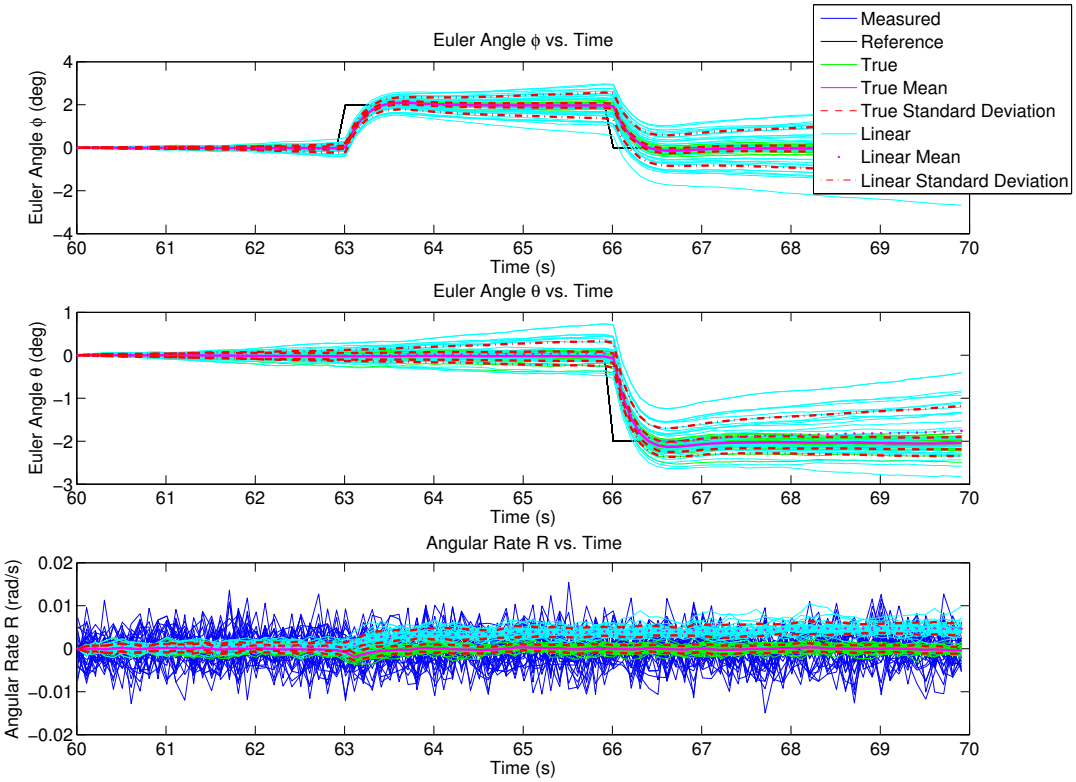

Figure 38. Linear Model 4 Dynamic Simulation Results with Second Modified Mixer Matrix.

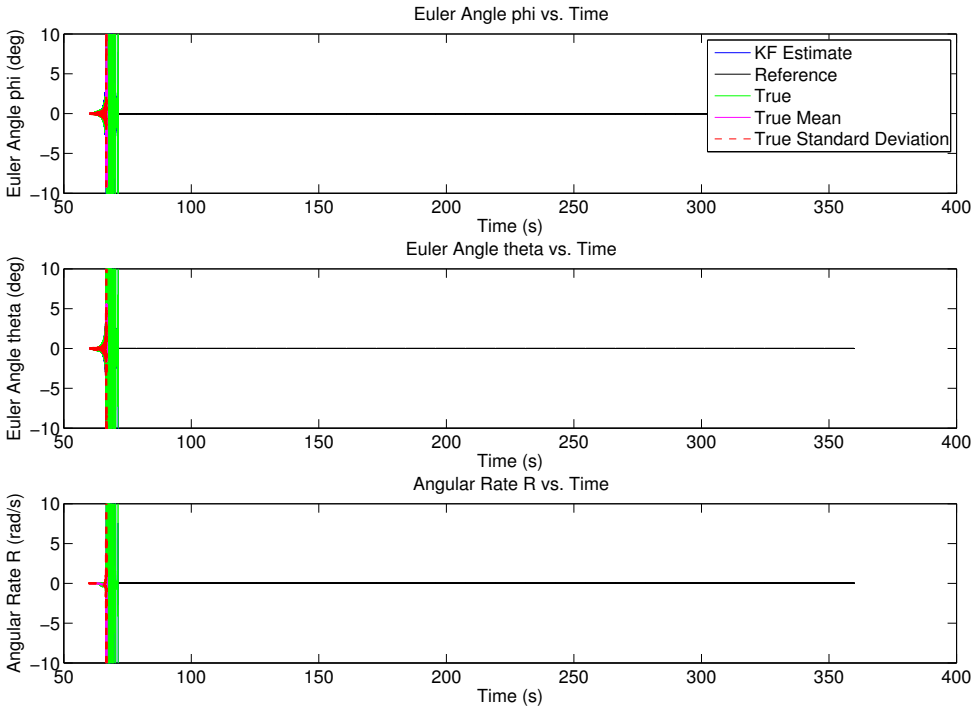

Figure 39. Linear Model 1 Static Simulation Results with Standard Mixer Matrix and Kalman Filter On.

for the plots shown in this section.

Figures 39 and 40 show the results of implementing the Kalman Filter based on the first linear model and comparing it to the simple non-linear model. It is immediately obvious that this filter quickly diverges which suggests a mismatch between the linear and non-linear model. Examination of Figures 29 and 30 show that the linear model may not be good enough basis for a Kalman Filter. In fact, substantially increasing the process noise matrix, which is equivalent to not relying on the linear model at all, yields results that are identical to the results shown in Figures 25 and 26. Since the plots are identical, they are not shown here. In contrast, the Kalman Filter based on the fourth linear model performs quite well. Figures 41 and 42 show the results. Unfortunately, there is little to no improvement over navigating off the raw measurements which can be seen by comparing the standard deviations in these figures

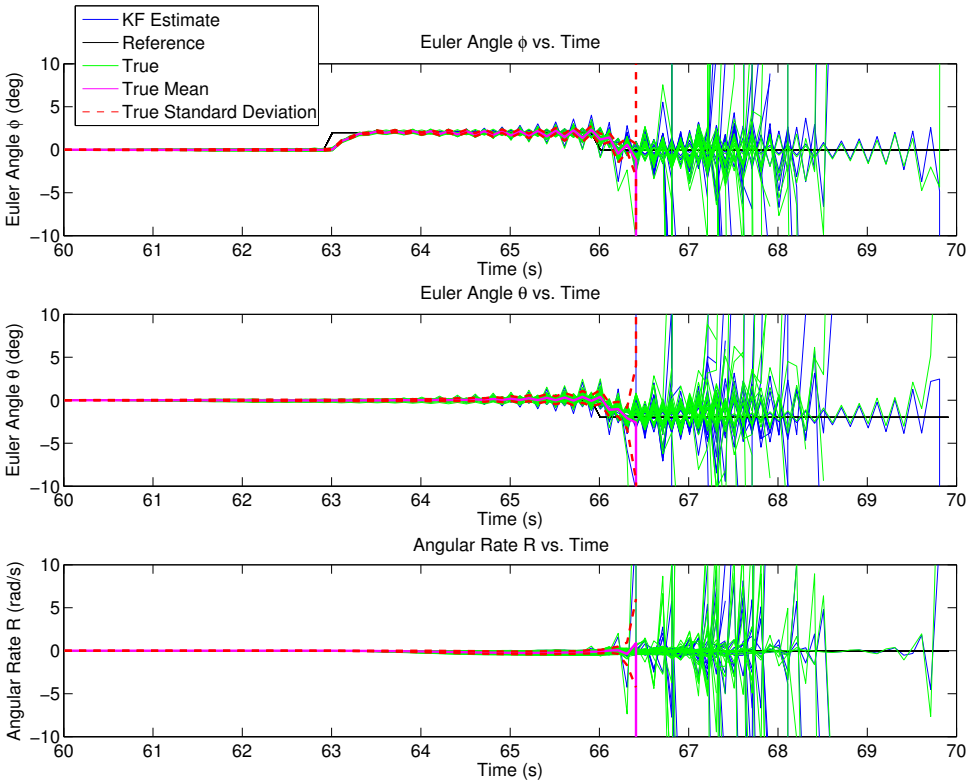

Figure 40. Linear Model 1 Dynamic Simulation Results with Standard Mixer Matrix and Kalman Filter On.

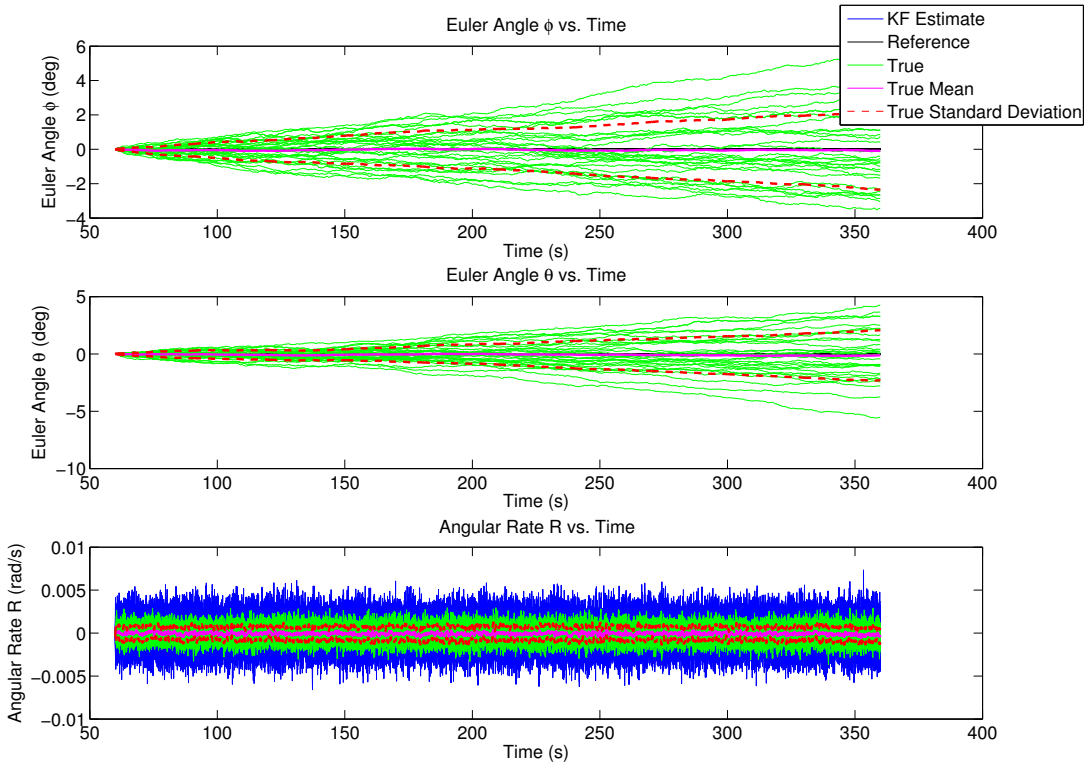

Figure 41. Linear Model 4 Static Simulation Results with Second Modified Mixer Matrix and Kalman Filter On.

to those in Figures 27 and 28. These results suggest that a Kalman Filter should not be used for this particular quadrotor.

5.5 Using Accelerometers to Determine the Gravity Vector

As mentioned in Section 2.3.2.2, the accelerometers can be used to measure the gravity vector if the quadrotor is moving at constant velocity, whether that velocity is zero or one-hundred feet per second. To examine this, some data was collected from the IMU while being held at a fixed height and position while rolling and pitching the IMU, called the static case. Data was also collected while moving the IMU around and rolling and pitching the IMU, called the dynamic case. In the static case where there are no accelerations, the roll and pitch angle can be determined from the readings of

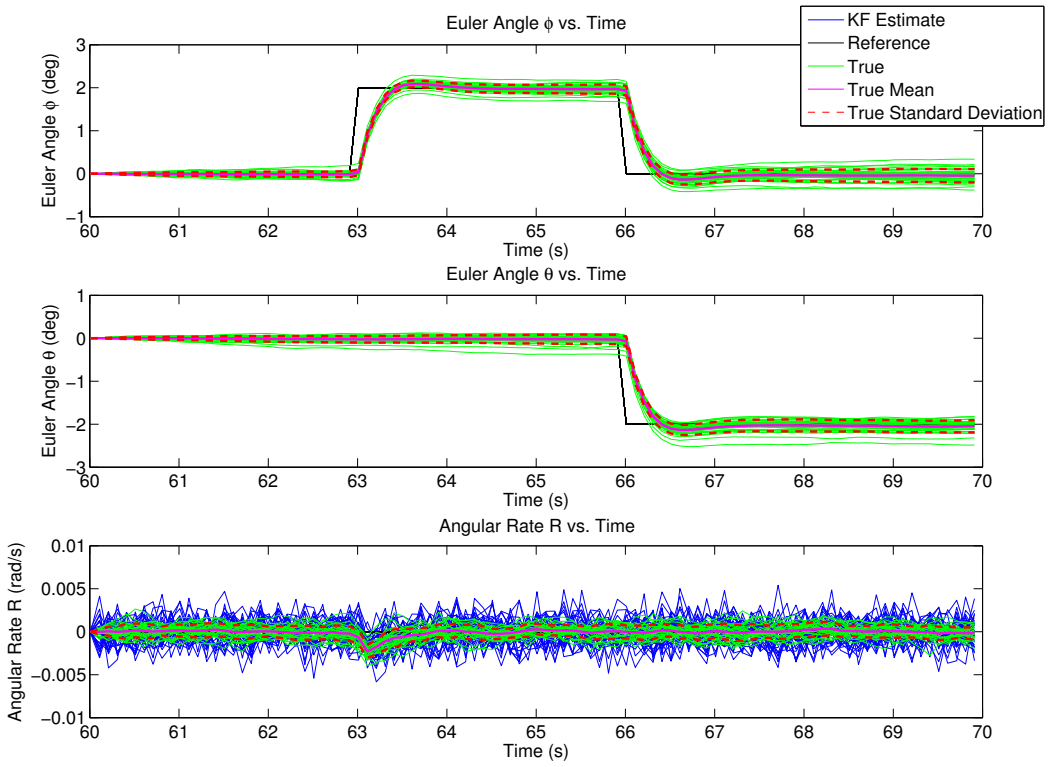

Figure 42. Linear Model 4 Dynamic Simulation Results with Second Modified Mixer Matrix and Kalman Filter On.

the x^b- and y^b-axis accelerometers. This is as simple as resolving the amount gravity that is directed along each of these two axes according to:

$$\begin{bmatrix} a_x^b \\ a_y^b \\ a_z^b \end{bmatrix} = C_n^b \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \tag{59}$$

By solving this equation for the roll and pitch angles, the roll and pitch angles are determined as shown:

$$\begin{bmatrix} a_x^b \\ a_y^b \\ a_z^b \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\theta\sin\phi & \sin\phi\sin\theta\sin\psi + \cos\theta\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}$$

$$\begin{bmatrix} a_x^b \\ a_y^b \\ a_z^b \end{bmatrix} = \begin{bmatrix} g\sin\theta \\ -g\sin\phi\cos\theta \\ -g\cos\phi\cos\theta \end{bmatrix}$$

$$\theta = \arcsin\left(\frac{a_x^b}{g}\right)$$

$$\phi = \arcsin\left(\frac{a_y^b}{-g\cos\theta}\right) \tag{60}$$

Unfortunately, this method does not allow determination of the yaw angle so a gyro must still be used to determine that angle.

Figure 43 shows the results of the static case and Figure 44 shows the results of the dynamic case. The first thing to note is that if the IMU is not moving and the roll and pitch are returned to zero, the angles determined from the accelerometer measurements always return to zero with some noise. In contrast, the gyro measurements, which are integrated to determine the angles, drift over time. The second thing to note is that when the IMU is subject to accelerations, the angles determined from the accelerometer measurements are unreliable. A possible method to correct long term

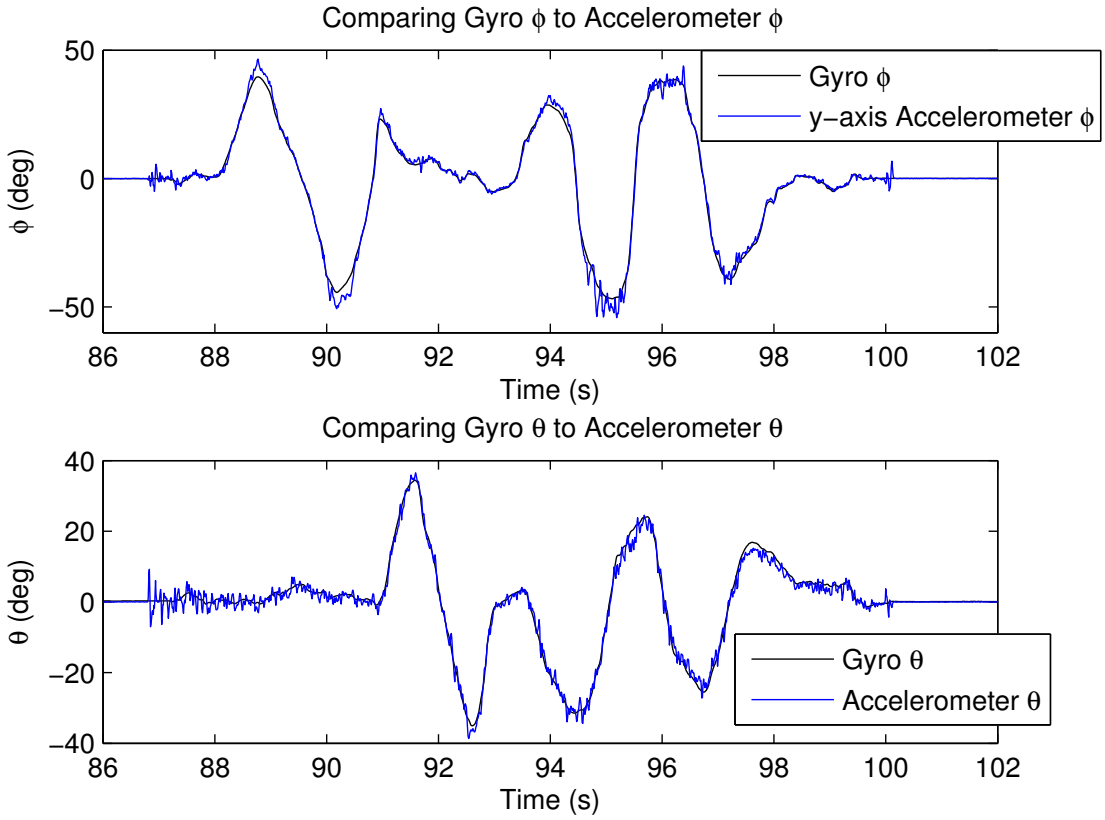

Figure 43. Using Accelerometers to Determine the Roll and Pitch Angles while Stationary.

gyro drift could include periodically landing the quadrotor to reset the gyros using the accelerometer measurements.

5.6 Using Accelerometers to Measure Angular Rates Results

As shown in Figure 5, the centripetal acceleration is $\omega^2 r$ where ω is the angular rate of the vehicle and r is the distance from the center of the quadrotor to where the accelerometers are located. The angular acceleration is α. The sum of all other external forces is grouped together as $\sum F$.

The difference can be taken between the pair of accelerometers that detect centripetal acceleration and between the pair of accelerometers that detect angular ac-

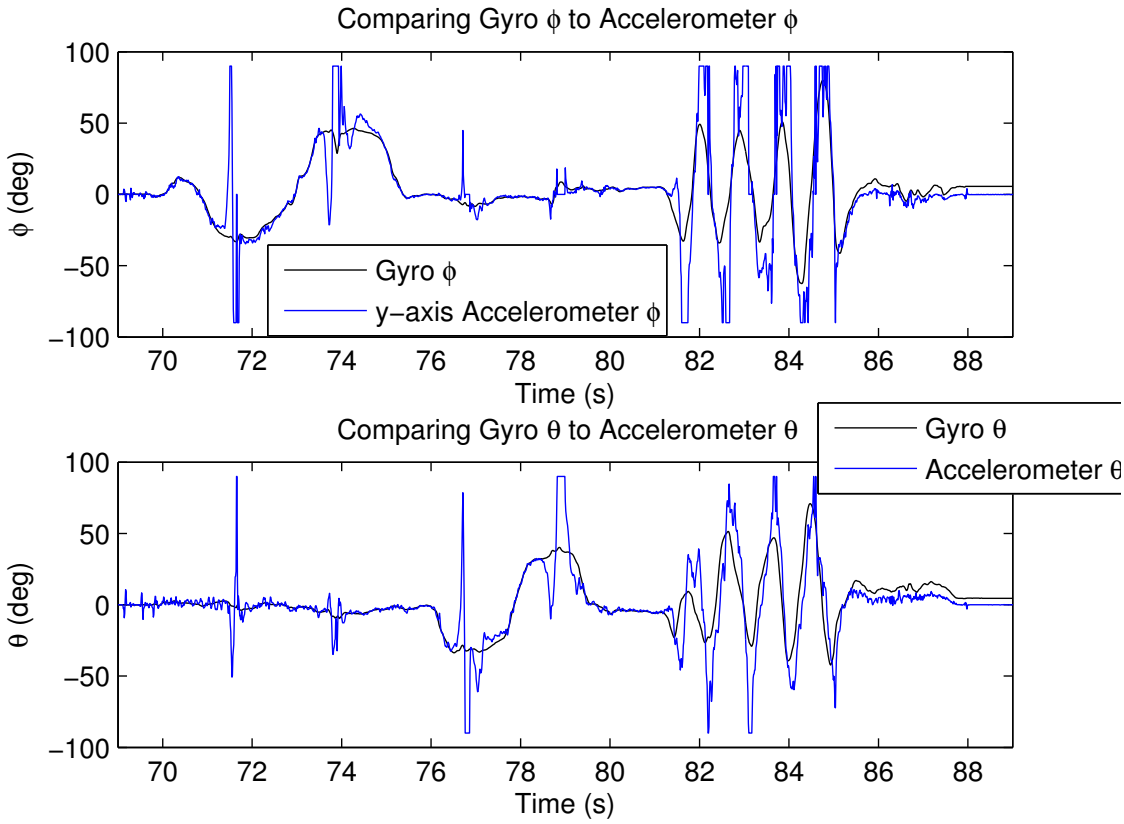

Figure 44. Using Accelerometers to Determine the Roll and Pitch Angles while Moving.

celeration in order to eliminate the external forces as shown:

$$\begin{aligned} a_{21} - a_{11} &= 2\omega^2 r \\ a_{12} - a_{22} &= 2\alpha r \end{aligned} \tag{61}$$

The angular rate and angular acceleration are then solved for as shown:

$$\begin{aligned} \omega &= \sqrt{\tfrac{a_{21}-a_{11}}{2r}} \\ \alpha &= \tfrac{a_{12}-a_{22}}{2r} \end{aligned} \tag{62}$$

Unfortunately, the centripetal acceleration method only captures the magnitude of the angular rate and not the sign. because of the square root relationship and the need to have a positive value under the square root. Additionally, the angular rate and angular acceleration do not have to be in the same direction (deceleration) so the sign of the measured angular acceleration cannot be used to determine the sign of the angular rate. However, the sign can be determined by using a gyro.

The error in these measurements can be determined by looking at the variance and standard deviation of these values. The standard deviation for the accelerometers and gyros are taken directly from the data sheet for the IMU. The standard deviation for the accelerometers is 0.005g and the standard deviation for the gyros is 0.2^o/s [18]. It is assumed that the accelerometer and gyro noise is zero mean, uncorrelated, white Gaussian noise. It is also assumed that all measurements are independent. The

variance is computed as:

$$
\begin{aligned}
&\text{Centripetal Acceleration} \\
E\left[\omega\omega^T\right] &= \frac{E\left[\sqrt{a_{21}-a_{11}}\sqrt{a_{21}-a_{11}}^T\right]}{2r} \\
E\left[\omega\omega^T\right] &= \frac{E\left[\sqrt{a_{21}a_{21}^T-a_{21}a_{11}^T-a_{11}a_{21}^T+a_{11}a_{11}^T}\right]}{2r} \\
&\text{Angular Acceleration} \\
E\left[\alpha\alpha^T\right] &= \frac{E\left[(a_{12}-a_{22})(a_{12}-a_{22})^T\right]}{4r^2} \\
E\left[\alpha\alpha^T\right] &= \frac{E\left[a_{12}a_{12}^T-a_{12}a_{22}^T-a_{22}a_{12}^T+a_{22}a_{22}^T\right]}{4r^2} \\
E\left[\alpha\alpha^T\right] &= \frac{E\left[a_{12}a_{12}^T\right]-E\left[a_{12}a_{22}^T\right]-E\left[a_{22}a_{12}^T\right]+E\left[a_{22}a_{22}^T\right]}{4r^2} \\
E\left[\alpha\alpha^T\right] &= \frac{E\left[a_{12}a_{12}^T\right]+E\left[a_{22}a_{22}^T\right]}{4r^2} \\
E\left[\alpha\alpha^T\right] &= \frac{2\sigma_a^2}{4r^2} \\
E\left[\alpha\alpha^T\right] &= \frac{\sigma_a^2}{2r^2}
\end{aligned}
\tag{63}
$$

where the E is the expectation operator.

Taking the expectation of a square root is a little more complex than desired for a simple error analysis, so the remainder of the error analysis will look at Monte-Carlo experiments. The Monte-Carlo experiments assume that $r = 1$ft and that $g = 32.2\frac{\text{ft}}{\text{s}^2}$. The Monte-Carlo experiments are each run for a time period of 10 minutes. The Monte-Carlo experiments using the centripetal acceleration method use gyro measurements to determine the sign of the angular rate. The gyro measurements have simulated noise with the same standard deviation as the IMU.

Figure 45 compares 30 Monte-Carlo experiments for the centripetal acceleration angle measurements and gyro angle measurements using the error statistics of the IMU and a simulated angular acceleration of $1^o/\text{s}^2$ for one second so that the angular rate from one second onward is $1^o/\text{s}$. Figure 46 shows a comparison of the angular acceleration angle measurements to the gyro angle measurements for the same experiment. In both figures, blue represents the gyro data, black the accelerometer data,

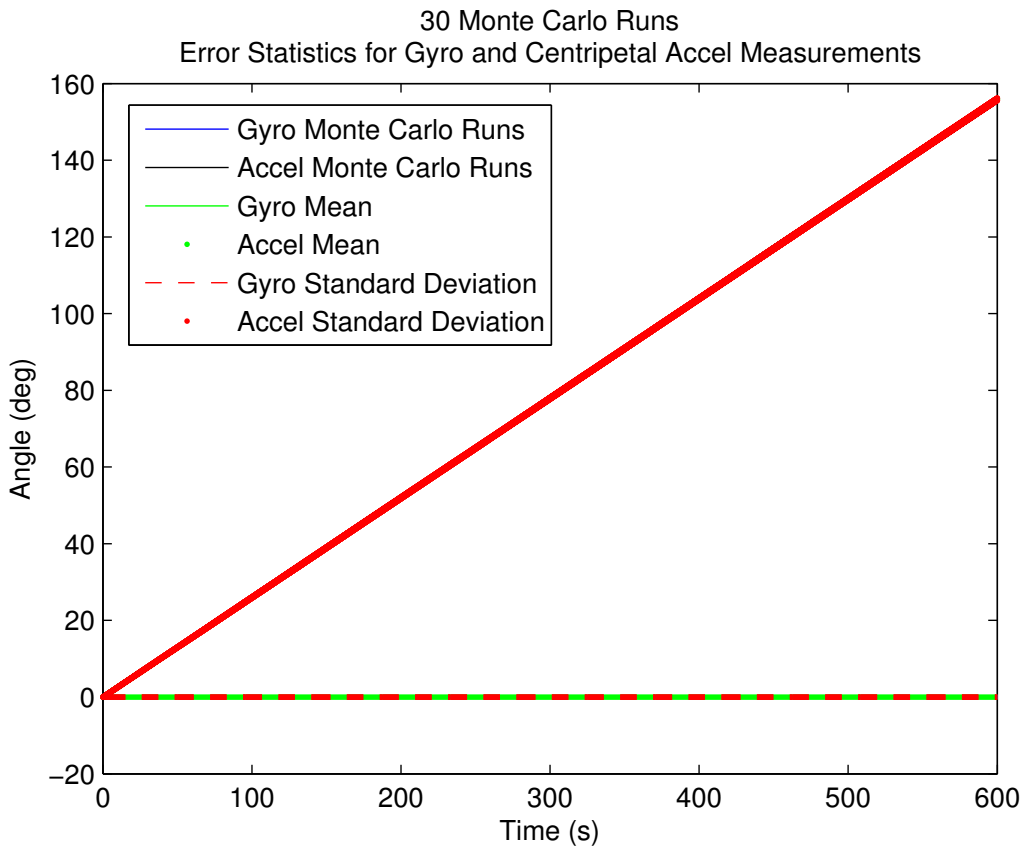

Figure 45. Centripetal Acceleration Method of Determining Angular rate vs Using Gyros with a simulated 1 degree per second angular rate and 5mg accelerometer noise.

green the mean, and red the standard deviation. The thicker green and red lines are for the accelerometers. Note that both figures show error between the gyro data and true, and between accelerometer data and true. In reality, the angle measured should grow linearly over time, which is why the gyro data always appears to be zero mean.

In Figure 45, because of the scale, the gyro data appears to be squashed to zero, but if we zoomed in, could see the usual gyro drift. The gyro data has zero mean error when compared to truth. The centripetal accelerometer remains close to zero mean since the angular rate is much lower than the noise floor of the accelerometer. When compared to truth, it appears to drift off by 160 degrees over the 10 minute period. Even though the angular rate is much lower than the noise floor, it biases the

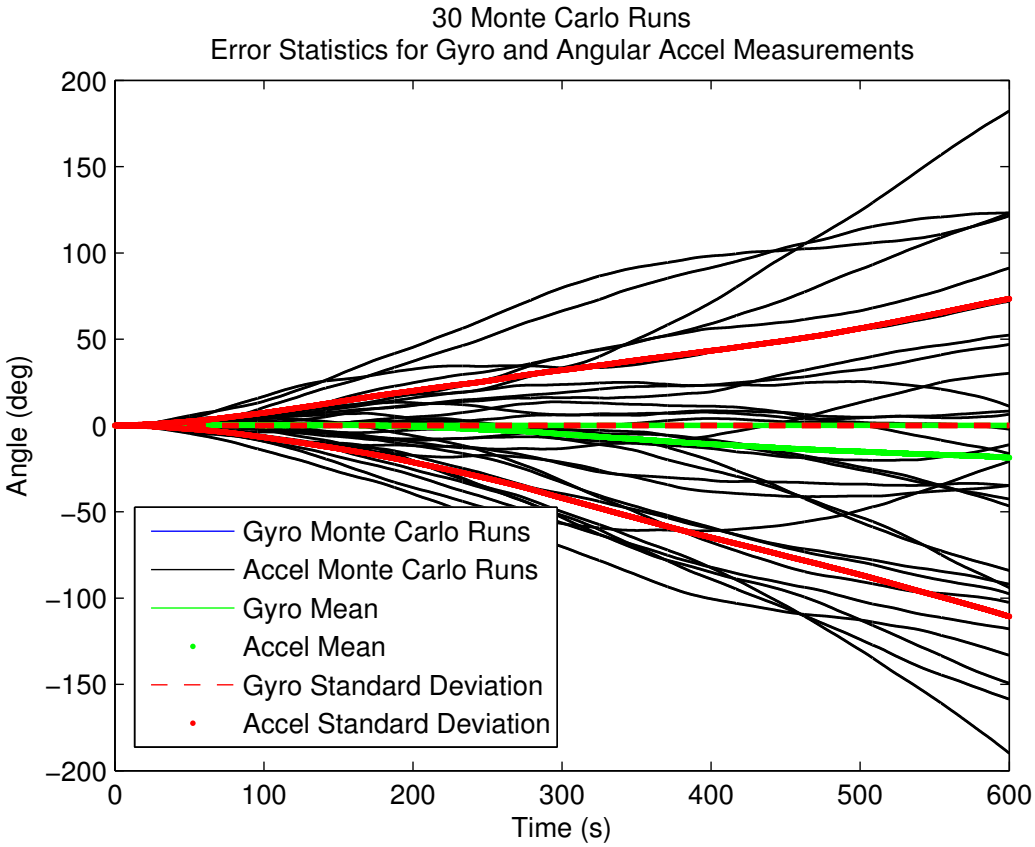

Figure 46. Centripetal Acceleration Method of Determining Angular rate vs Using Gyros with a simulated 1 degree per second angular rate and 5mg accelerometer noise.

accelerometer measurements enough that it picks up some of the angular rate, but not all of it.

On the other hand, the angular accelerometers, while showing little drift from truth in Figure 46, has a terrible standard deviation which grows quadratically with time since there are two integrations.

If the angular rate is above the noise floor significantly enough, the centripetal acceleration method works very well, however with error growth substantially larger than the gyro measurements. Unfortunately, with the 5mg accelerometer technology, this angular rate needs to be approximately $300^o/\text{s}$, setting a lower bound on the dynamics at which the centripetal acceleration method can be used.

The $300^o/\text{s}$ is reached by simulating a $300^o/\text{s}^2$ angular acceleration for one second. Figure 47 shows how the centripetal acceleration method would perform under these conditions. Note that there is no significant change to the error seen when using the angular acceleration method as shown in Figure 47 since the noise threshold is not a big factor for the angular acceleration method.

Now, suppose that accelerometers only one order of magnitude better were available. These accelerometers would have a $500\mu\text{g}$ noise standard deviation. If these accelerometers were available, the angular rate threshold drops to approximately $100^o/\text{s}$. The $100^o/\text{s}$ is reached by simulating a $100^o/\text{s}^2$ angular acceleration for one second. Figures 49 and 50 show the performance of the centripetal and angular acceleration methods respectively. As expected, using better accelerometers results in improved performance. At this point, the centripetal accelerometer method roughly matches using the gyro measurements.

This trend of getting improved performance and a lower threshold on the angular rate using the centripetal acceleration method continues, but with diminishing returns. This is due to the fact that the centripetal acceleration still requires the use of

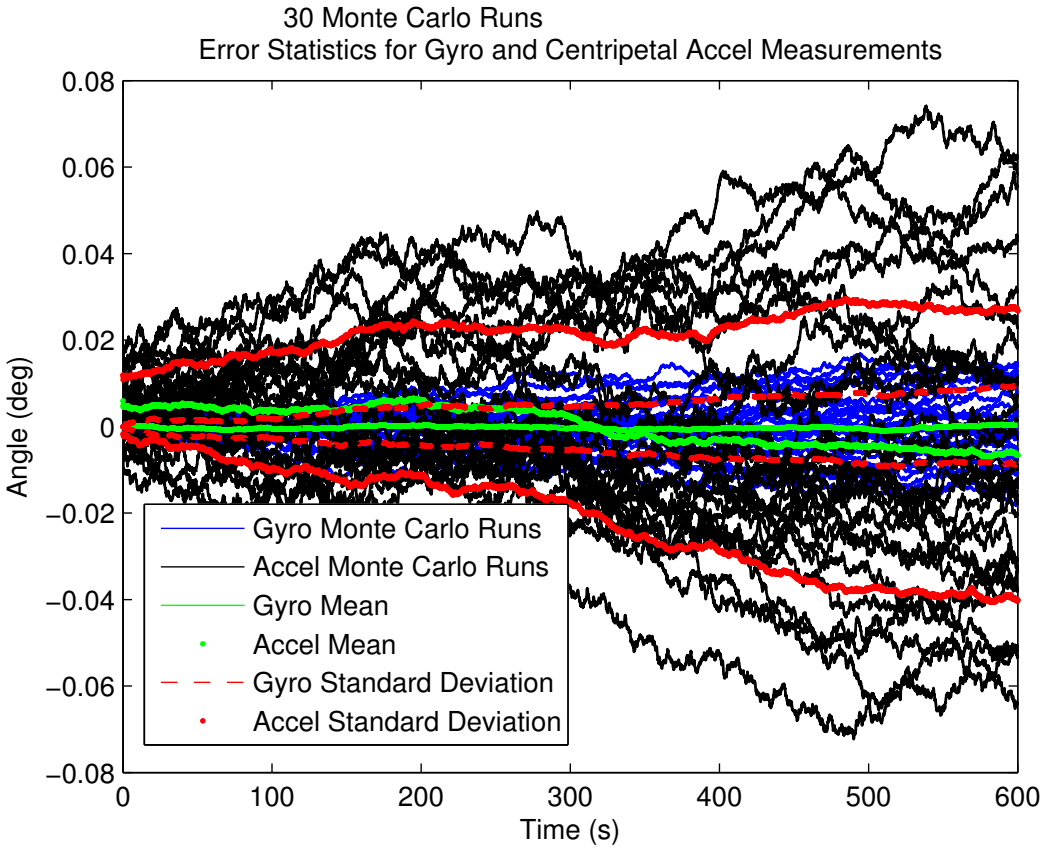

Figure 47. Centripetal Acceleration Method of Determining Angular rate vs Using Gyros with a simulated 300 degree per second angular rate and 5mg accelerometer noise.

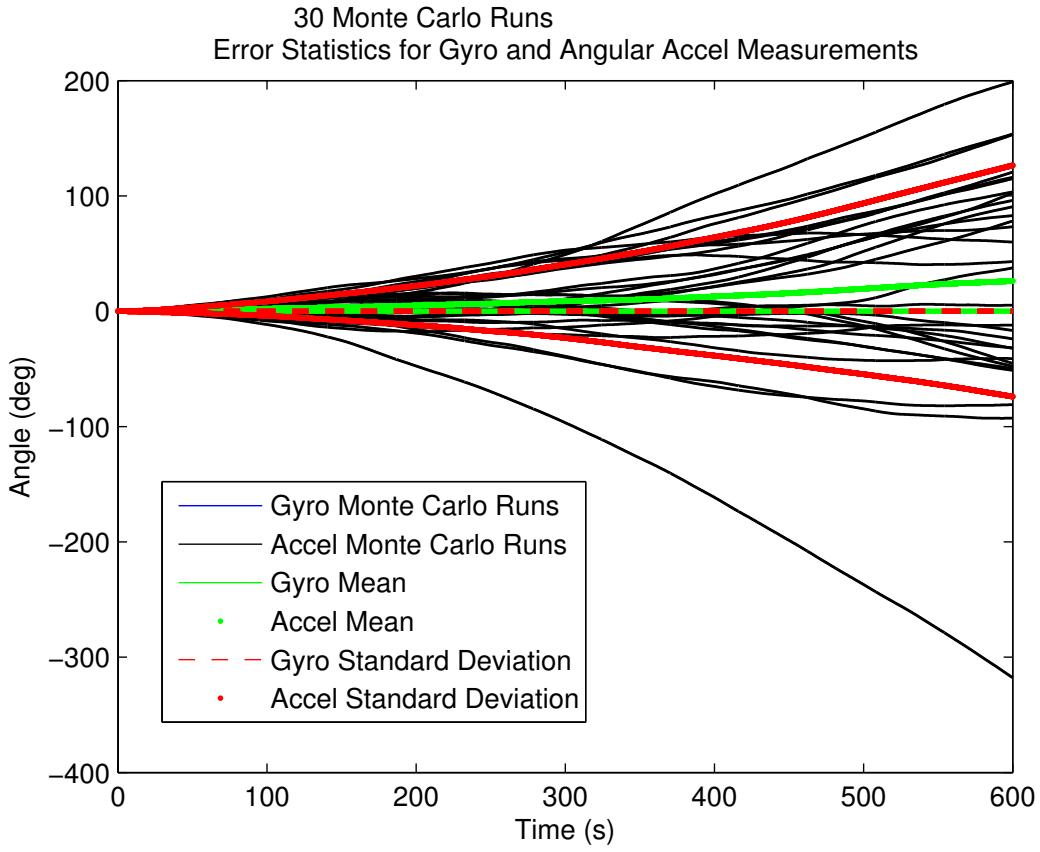

Figure 48. Centripetal Acceleration Method of Determining Angular rate vs Using Gyros with a simulated 300 degree per second angular rate and 5mg accelerometer noise.

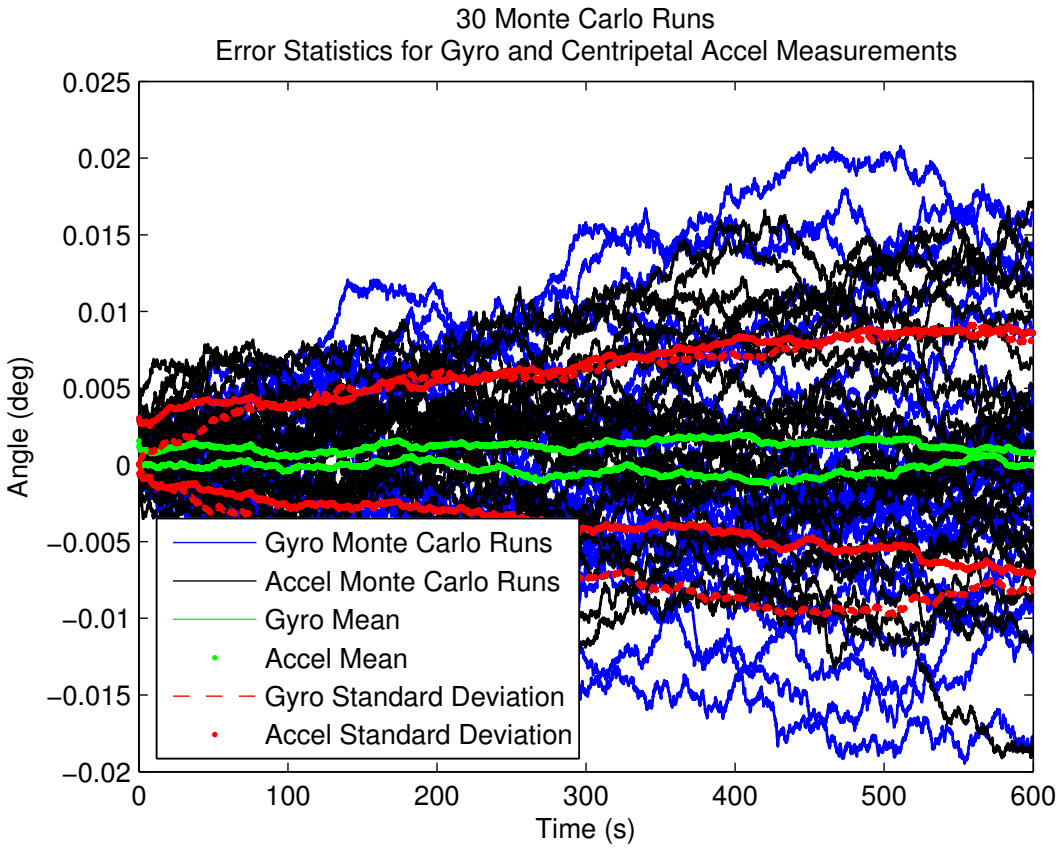

Figure 49. Centripetal Acceleration Method of Determining Angular rate vs Using Gyros with a simulated 100 degree per second angular rate and 500ug accelerometer noise.

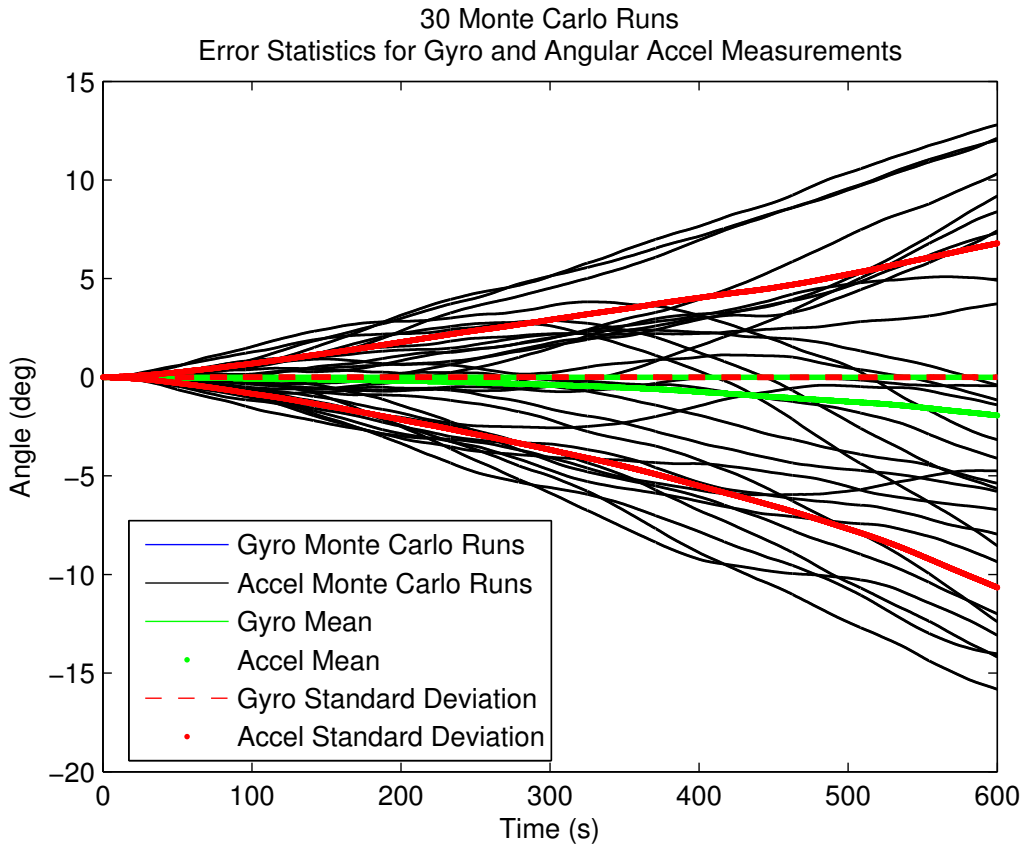

Figure 50. Centripetal Acceleration Method of Determining Angular rate vs Using Gyros with a simulated 100 degree per second angular rate and 500ug accelerometer noise.

a gyro to determine the sign of the angular rate. As the angular rate approaches the gyro noise floor, improving the accelerometer technology further does little to improve upon the angular rate threshold. As a result, it would be better to use some sort of switch that, if the angular rate is below the threshold, the gyro is used to navigate, and if the angular rate is above the threshold, the centripetal accelerometers are used to navigate.

A summary of the accelerometer noise standard deviation and the associated angular rate threshold along with existing accelerometer technologies which can achieve that performance is shown in Table 3. Figure 51 shows the performance of the centripetal acceleration method if seismic accelerometers are used. Figure 52 shows the performance of the angular acceleration method for the same.

Interestingly, for the accuracy of the seismic accelerometers, the angular acceleration method outperforms the centripetal acceleration method, at least over the first ten minutes. However, ultimately, because of quadratic error growth vs. linear error growth, the angular acceleration method will do worse after enough time has passed. Unfortunately, the limiting factor on the more accurate accelerometers is their dynamic range. The seismic accelerometers in particular have a ± 0.5g range, which limits the maximum detectable angular rate using the centripetal acceleration method to approximately $230^o/\text{s}$ with $r = 1$ft still. This limitation does not have nearly the same impact on the angular acceleration method which would be limited to detecting maximum accelerations of approximately $1850^o/\text{s}^2$.

5.7 Periodic Landing to Improve Attitude Solution

The results of Section 5.5 suggest a possible solution to the long term drift in attitude. It may be possible to periodically land and while landed, measure the pitch and roll angle from the accelermoeters according to Equation 60. At the same time,

Table 3. Accelerometer Noise Standard Deviation vs. Angular Rate Threshold and Accelerometer Technology

Accelerometer Noise	Angular Rate Threshold	Technology
5mg	300^o/s	3DM-GX3-25 [18]
500μg	100^o/s	
50μg	30^o/s	INN-202 [3], A40 [10], and CXL02TG3 [17]
5μg	5^o/s	
500ng	2^o/s	
50ng	1^o/s	
5ng	1^o/s	Model 86 [7] and Model 731A [21]

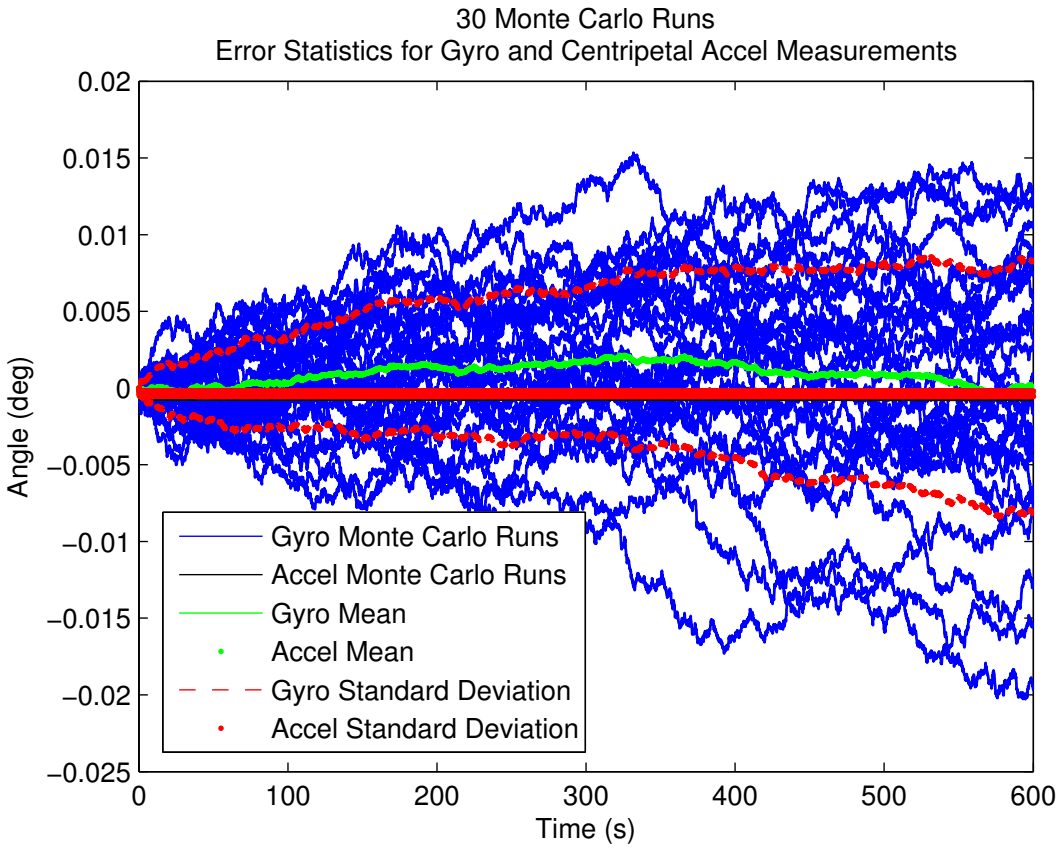

Figure 51. Centripetal Acceleration Method of Determining Angular rate vs Using Gyros with a simulated 1 degree per second angular rate and 5ng accelerometer noise.

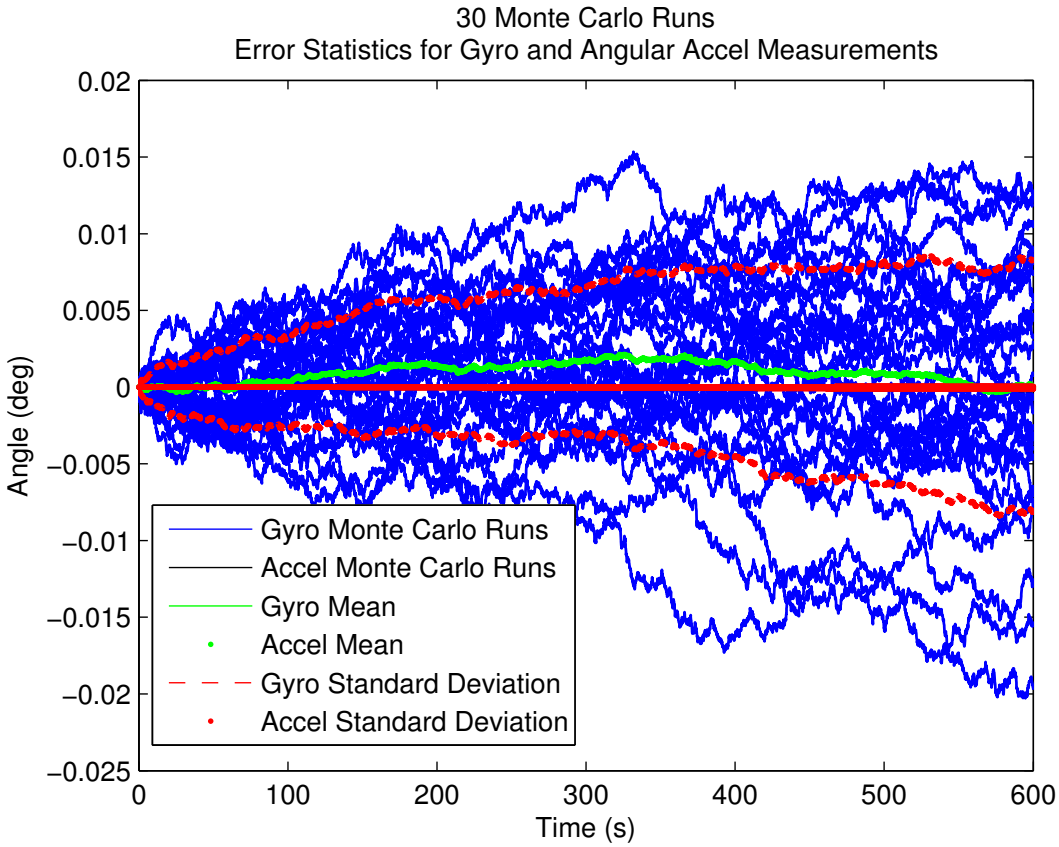

Figure 52. Angular Acceleration Method of Determining Angular rate vs Using Gyros with a simulated 1 degree per second angular rate and 5ng accelerometer noise.

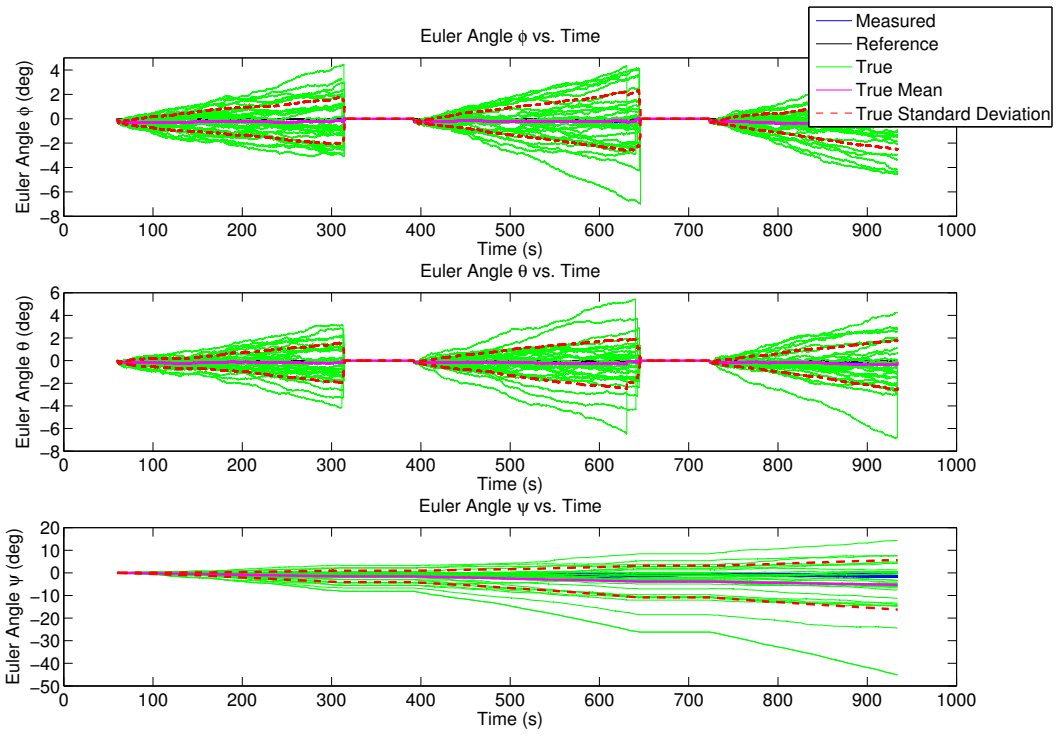

Figure 53. Periodically Landing Quadrotor on Flat Plane.

the bias in the gyro measurements could be recalculated to account for any long term drift in the measured angular rates. In order to reduce noise in the accelerometer measurements, a time average of the accelerometer measurements over the time the quadrotor is landed is used to determine the roll and pitch angles rather than using the raw measurements.

Figure 53 shows exactly how that would look for a flight profile which causes the quadrotor to lift off and attempt to hover, land at some predetermined point in time, reset the roll and pitch and recalculate the bias while landed, and then take off again. This cycle is then repeated. In this figure, the simulated profile has the quadrotor landing on a flat plane.

This can be compared to a flight profile where the quadrotor does not land and the gyro measurements drift significantly over time as shown in Figure 54. It is clearly

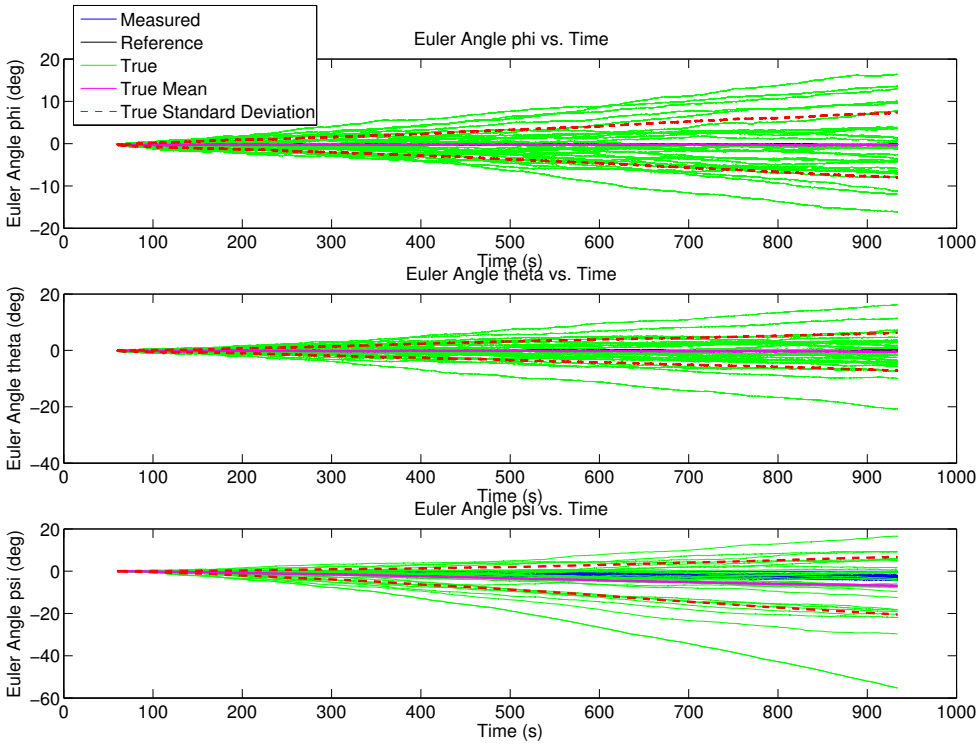

Figure 54. Quadrotor Attempting to Hover.

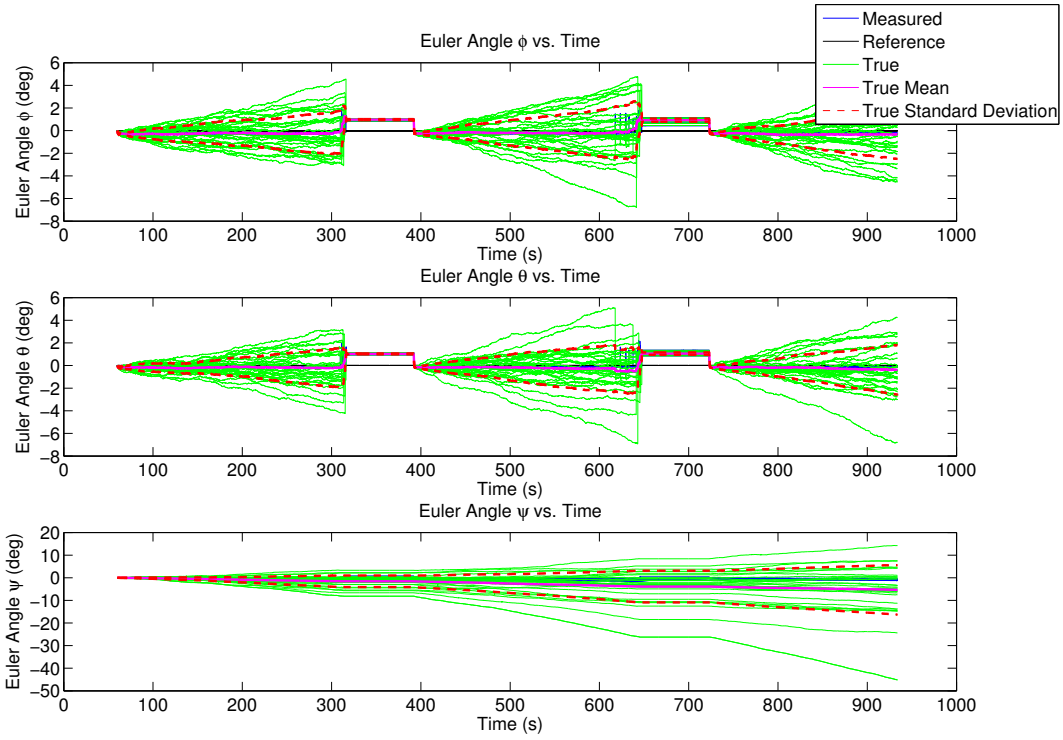

Figure 55. Periodically Landing Quadrotor on an Inclined Plane.

evident that landing once approximately every four minutes constrains the standard deviation to approximately two degrees while not landing allows the standard deviation to grow to ten degrees over the length of the simulation. Also of note is that periodic landing does not constrain the yaw since yaw is not observable when measuring the gravity vector. This method is also shown to work for inclined planes. Figure 55 shows the results of landing on a plane that is rotated one degree in both the roll and pitch. Since the command inputs are for a hover, the quadrotor quickly returns to 0 degrees roll and 0 degrees pitch after take off.

Of note in these simulation is that the landing is not realistic. The purpose of this simulation was to show how the attitude could be bounded in the long term rather than drifting off, not to simulate a realistic landing. Rotor ground effect and bouncing upon touch down are completely ignored. Instead, a simple check is

performed to determine if the quadrotor is at or below the plane representing the ground in order to tell whether the quadrotor is flying or has landed. The method of landing the quadrotor in simulation was simply allowing the quadrotor to accelerate downward until it hit the ground, at which point it was considered to have landed. This is also why it appears that the quadrotor does not always land at the exact same point in time since different noise profiles result in different flight trajectories while attempting to maintain a hover before landing.

VI. Conclusion

Major achievements of this thesis include the development of non-linear 6 degree-of-freedom quadrotor models, development of linear models to approximate the non-linear model, and a simulator to test and compare these models. The 6 degree-of-freedom quadrotor simulator represents a substantial contribution to the Advanced Navigation and Technology (ANT) Center since it can be easily modified to simulate any quadrotor if its mass, moments of inertia, center of gravity, and motor locations relative to the center of gravity are known. If at least one of the non-linear models can be verified in flight test, the simulator can be used to evaluate the performance of existing and future quadrotor designs. It could also be used to predict quadrotor performance if other sensors are integrated into the quadrotor and the measurement equations for those sensors are incorporated into the simulator.

One of the linear models was successfully used in a Kalman Filter, but did not offer improved performance over navigating from IMU measurements alone. This model was improvised and would require re-derivation for each new quadrotor configuration since it changes depending on the center of gravity. Unfortunately, the linear models could not be validated against the non-linear models. The linear models generated in this thesis do not appear to be the best solution to increasing long term quadrotor stability. It may be possible to determine a better mixer matrix solution that works for all quadrotor simulators, or it may be possible to find a solution in the control loops.

One way to achieve better long term stability was discovered. It requires the quadrotor to land periodically and reset the bias on the gyros by using the accelerometers to determine the gravity vector. It was shown that landing about every four minutes limits the standard deviation of the roll and pitch angle drift to approximately 2 degrees. Flight profiles like these might represent the quadrotor taking off

and flying to one or more target locations, landing and observing some target at the target locations, then returning to its take off point. Since the quadrotor in this thesis has a flight time of approximately 20 minutes, flight profiles like these could result in missions that are several hours long, or at least as long as the battery powering the FPGA, IMU, and radio receiver allows, but only require 20 total minutes of flight time. This flight time could be improved if more efficient motors or better batteries could be developed. This flight time can also be extended if periodic landing is incorporated with the ANT Center and Air Force Research Laboratory project to land a hovering vehicle on a power line in order to harvest power. At the same time the quadrotor is recharging it's batteries, it can also reset it's roll and pitch angle. If the power line has been surveyed and it's direction is well known, the quadrotor could potentially reset its yaw angle as well.

It is left as future work to determine how long the quadrotor must land in order to get a good roll and pitch measurement from the accelerometers and to re-calculate the bias in the gyros. It is also important to note that this requires no improvement in IMU technology. Better IMU technology would simply mean that the quadrotor could fly for longer time periods betweeen landing. This technique can be extended to work on any hovering vehicle. It could also be used on fixed wing aircraft, but instead of landing, this method could be used when flying wings level at constant velocity.

Another potential way performance could be improved includes using accelerometers to measure angular accelerations and angular rates as suggested in Section 5.6. The limiting factor is the current available technology. If seismic accelerometers can be improved to have a greater dynamic range, they may be able to replace the gyro as the primary inertial sensor to determine attitude and use gyros to augment them. Accelerometers based on cold atom technology may fill this role in the near future.

Characterization of the measurement noise in order to create an accurate measurement model gave mixed results. While some of the gyros and accelerometers were easy to model, some were not. It is possible that estimating more than the random walk and bias instability to match the Allan Variance plots in Figures 13 and 14 could result in a better noise characterization, but is left for future work. It is also possible that another Microstrain 3DM-GX3-25 might have measurement noise characteristics that are easier to model simply due to manufacturing inconsistencies.

Bibliography

[1] Balas, C. *Modelling and Linear Control of a Quadrotor*. Master's thesis, Cranfield University, 2007.

[2] Bhargava, Abhishek. *Development of a Quadrotor Testbed for Control and Sensor Development*. Master's thesis, Clemson University, 2008.

[3] Bricett. *Quartz Accelerometer INN-202 Datasheet*. Technical report, Bricett, 2010.

[4] Chen, Tsung-Lin. "Design and Analysis of a Fault-Tolerant Coplanar Gyro-Free Inertial Measurement Unit". *JOURNAL OF MICROELECTROMECHANICAL SYSTEMS*, 17:201–212, 2008.

[5] Cowling, Ian. *Towards Autonomy of a Quadrotor UAV*. Ph.D. thesis, Cranfield University, 2008.

[6] El-Sheimy, Naser, Haiying Hou, and Xiaoji Niu. "Analysis and Modeling of Inertial Sensors Using Allan Variance". *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, 57:140–149, 2008.

[7] Endevco. *Model 86 Seismic Accelerometer*. Technical report, Endevco.

[8] Finney, Ross L., Maurice D. Wier, and Frank R. Giordano. *Thomas' Calculus Early Transcendentals*. Addison-Wesley, updated tenth edition edition, 2003.

[9] Flowers, Spencer G. *Stabilization and Control of a Quad-Rotor Micro-UAV Using Vision Sensors*. Master's thesis, Brigham Young University, 2008.

[10] Gladiator Technologies, Inc. *A40 MEMS Acc*. Technical report, Gladiator Technologies, Inc., 2010.

[11] Huang, Haomiao, Gabriel M. Hoffmann, Steven L. Waslander, and Claire J. Tomlin. "Aerodynamics and Control of Autonomous Quadrotor Helicopters in Aggressive Maneuvering". *IEEE International Conference on Robotics and Automation*, 3277–3282, 2009.

[12] Johnson, Neil G. *Vision-Assisted Control of a Hovering Air Vehicle in an Indoor Setting*. Master's thesis, Brigham Young University, 2008.

[13] Kim, Hyunseok, Jang Gyu Lee, and Chan Gook Park. "Performance Improvement of GPS/INS Integrated System Using Allan Variance Analysis". *The 2004 International Symposium on GNSS/GPS*. 2004.

[14] Labadille, Muriel. *Non-linear Control of a Quadrotor*. Master's thesis, Cranfield University, 2007.

[15] Martin, Philippe and Erwan Salaün. "The True Role of Accelerometer Feedback in Quadrotor Control". *IEEE International Conference on Robotics and Automation*, 1623–1629, 2010.

[16] Maybeck, Peter S. *Stochastic Models, Estimation, and Control Volume 1*. Navtech Book & Software Store, Arlington, VA, 1994.

[17] MEMSIC. *CXLTG-Series High Performance Accelerometer Datasheet*. Technical report, MEMSIC Inc.

[18] Microstrain. *Microstrain 3DM-GX3-25 Data Sheet*. Technical report, Microstrain, 2009.

[19] Munson, Kenneth. *Helicopters and Other Rotary Aircraft Since 1907*. The Macmillan Company - New York, first american edition edition, 1969.

[20] Ng, Lawrence C. *On the Application of Allan Variance for Ring Laser Gyro Performance Characterization*. Technical report, Lawrence Livermore National Laboratory, 1993.

[21] Research, Wilcoxon. *Model 731A/Model P31 System Seismic Accelerometer and Power Amplifier*. Technical report, Wilcoxon Research, 1996.

[22] Shepherd, Jack F. *A Hierarchical Neuro-Evolutionary Approach to Small Quadrotor Control*. Master's thesis, Oregon State University, 2009.

[23] Stepaniak, Michael J. *A Quadrotor Sensor Platform*. Ph.D. thesis, Ohio University, 2008.

[24] Tan, Chin-Woo and Sungsu Park. "Design of Accelerometer-Based Inertial Navigation Systems". *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, 54:2520–2530, 2005.

[25] Tayebi, Abdelhamid and Stephen McGilvray. "Attitude Stabilization of a VTOL Quadrotor Aircraft". *IEEE Transactions on Control Systems Technology*, 14:562–571, 2006.

[26] Titterton, David and John Weston. *Strapdown Inertial Navigation Technology*. The Institution of Engineering and Technology, London, United Kingdom and The American Institute of Aeronautics, Reston, Virginia, USA, 2nd edition, 2004.

[27] Woodman, Oliver J. *An Introduction to Inertial Navigation*. Technical report, University of Cambridge Computer Laboratory, 2007.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704–0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
24–03–2010	Master's Thesis	Aug 2009 — Mar 2011

4. TITLE AND SUBTITLE

LONG TERM QUADROTOR STABILIZATION

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Nicolas Hamilton, 2Lt, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/11-13

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

"Intentionally left blank."

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

"This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States."

14. ABSTRACT

The work of this thesis focuses on the IMU and getting the best performance possible out of the IMU to achieve better long term stability and a better navigation solution. This is done in two ways. First, the IMU accelerometer output is examined to determine if it is possible to use accelerometers to determine attitude. If the quadrotor is stationary or moving at constant velocity, the roll and pitch angles can be determined. Additionally, the accelerometers can be used to determine angular accelerations and angular rates which are integrated to determine heading. The second approach models the quadrotor and uses the models in Kalman Filters along with the IMU measurements to determine the best possible navigation solution.

15. SUBJECT TERMS

Quadrotor, Inertial Measurement Unit

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Michael J. Stepaniak, Lt Col, USAF, (ENG)
U	U	U	UU	118	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636, x4603; michael.stepaniak@afit.edu