SYSTEM-LEVEL ANALYSIS OF AUTONOMOUS UAV LANDING SENSITIVITIES IN GPS-DENIED ENVIRONMENTS

by

Terran R. Gerratt

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Electrical Engineering

Approved:

Randall Christensen, Ph.D. Major Professor	Greg Droge, Ph.D. Committee Member
Douglas Hunsaker, Ph.D. Committee Member	Richard S. Inouye, Ph.D. Vice Provost for Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2020

ABSTRACT

System-Level Analysis of Autonomous UAV Landing Sensitivities in GPS-denied Environments

by

Terran R. Gerratt, Master of Science

Utah State University, 2020

Major Professor: Randall Christensen, Ph.D.
Department: Electrical and Computer Engineering

This paper presents a system-level analysis of navigation errors in an autonomous UAV carrier-landing scenario with an emphasis on fixed-wing aircraft. The paper focuses on typical sensors used in UAV landings which include an IMU, GPS, camera, and lidar. To accurately model the carrier's behavior, data from the Office of Naval Research SCONE project is used to represent a typical carrier that would be used in a landing scenario.

The analysis is accomplished using a Monte Carlo simulation and an extended Kalman filter to estimate the position, velocity, and attitude of both UAV and carrier. Particular interest is placed on the relative state covariances between the two vehicles three seconds prior to landing. The landing system's sensitivity to onboard sensor errors, GPS signal dependence, and passive fiducial numbers are analyzed by varying sensor grade and position measurement availability. The results of the analysis enable the development of less expensive landing systems without compromising safety or accuracy.

(117 pages)

PUBLIC ABSTRACT

System-Level Analysis of Autonomous UAV Landing Sensitivities in GPS-denied Environments

Terran R. Gerratt

This paper presents an analysis of the navigation accuracy of an fixed-wing Unmanned Aerial Vehicle (UAV) landing on a aircraft carrier. The UAV is equipped with typical sensors used in landing scenarios. Data from the Office of Naval Research is used to accurately capture the behavior of the aircraft carrier. Through simulation, the position and orientation of both the UAV and carrier are estimated. The quality of the UAV's sensors are varied to determine the sensitivity of these estimates to sensor accuracy. The system's sensitivity to GPS signals and visual markers on the carrier is also analyzed. These results allow designers to choose the most economical sensors for landing systems that provide a safe and accurate landing.

To my ever supportive and patient wife, Alexis

ACKNOWLEDGMENTS

This work would not have been possible without the countless hours of help and direction from my professor, Dr. Randall Christensen. I would also like to thank my committee, Dr. Greg Droge and Dr. Douglas Hunsaker for their valuable feedback and insights.

Finally, I would like to thank my beautiful wife, Alexis, for always believing in me and pushing me to reach my goals.

Terran R. Gerratt

CONTENTS

Page

ABSTRACT . . . iii

PUBLIC ABSTRACT . . . iv

ACKNOWLEDGMENTS . . . vi

LIST OF TABLES . . . ix

LIST OF FIGURES . . . x

ACRONYMS . . . xii

1 INTRODUCTION . . . 1
- 1.1 Objectives . . . 1
- 1.2 Overview . . . 2

2 LITERATURE REVIEW AND BACKGROUND . . . 3
- 2.1 Offboard Sensors . . . 3
- 2.2 Passive Fiducials . . . 4
- 2.3 Computer-Vision Tracking . . . 5
- 2.4 Lidar-Aided Relative Navigation . . . 5

3 MATERIALS AND METHODS . . . 7
- 3.1 Notation and Conventions . . . 7
- 3.2 Sensor Suite . . . 9
- 3.3 Carrier Model . . . 11
- 3.4 Estimation and Simulation . . . 11

4 STATE DEFINITIONS . . . 13
- 4.1 Coordinate Frames . . . 13
- 4.2 State Vectors . . . 14
- 4.3 Error Definitions . . . 16

5 NAVIGATION AND STATE ESTIMATION . . . 20
- 5.1 Nonlinear Truth State Propagation . . . 20
- 5.2 Nonlinear Navigation Differential Equations . . . 22
- 5.3 Nonlinear Measurement Models . . . 25
- 5.4 Linearized Design Model . . . 27
- 5.5 Linearized Measurement Model . . . 32
- 5.6 Covariance Propagation . . . 33
- 5.7 Kalman Update . . . 35
- 5.8 Relative Covariance Transforms . . . 36

6 FILTER VALIDATION . . . 37
6.1 Error State Vector Validation . . . 37
6.2 Nonlinear State Propagation . . . 38
6.3 Simulation Parameters . . . 41
6.4 Residual Verification . . . 45
6.5 Covariance Propagation . . . 47
6.6 State Estimation . . . 52

7 RESULTS . . . 55
7.1 Sensor Grade Sensitivity . . . 55
7.1.1 IMU Grade . . . 57
7.1.2 Camera Errors . . . 59
7.1.3 Lidar Errors . . . 59
7.2 Environmental Sensitivity . . . 62
7.2.1 GPS-denied . . . 62
7.2.2 Fiducial Number . . . 62

8 CONCLUSION AND FUTURE WORK . . . 65
8.1 Conclusion . . . 65
8.2 Future Work . . . 66

REFERENCES . . . 67

APPENDICES . . . 70
A Error State Linearization . . . 71
A.1 Velocity Linearization . . . 71
A.2 Quaternion Linearization . . . 72
A.3 Linearized F Matrix . . . 75
A.4 Linearization Validation . . . 76
B Measurement Model Linearization . . . 78
B.1 Line-of-Sight Linearization . . . 78
B.2 Lidar Linearization . . . 81
B.3 GPS Linearization . . . 83
B.4 Linearization Validation . . . 84
C Filter Validation . . . 86
C.1 Nonlinear State Propagation . . . 86
C.2 Covarinace Propagation . . . 87
C.3 State Estimation . . . 90
D Results . . . 96
D.1 IMU Grade . . . 96
D.2 Camera Errors . . . 98
D.3 Lidar Errors . . . 100
D.4 GPS-denied . . . 102
D.5 Fiducial Number . . . 104

LIST OF TABLES

Table Page

4.1 State variable descriptions 15

6.1 Mapping equation validation 38

6.2 Measurement parameters 43

6.3 Accelerometer parameters 43

6.4 Gyroscope parameters 43

6.5 Carrier parameters 44

6.6 State initial uncertainty values 44

6.7 State estimation parameters 52

7.1 Sensor grade specifications 56

A.1 Linear system validation 77

B.1 Linear measurement validation 85

LIST OF FIGURES

Figure Page

3.1 Use of lidar measurement to determine carrier attitude 10

4.1 Coordinate frame diagram 14

6.1 UAV flight trajectory 39

6.2 Camera residual propagation 40

6.3 Lidar residual propagation 40

6.4 GPS residual propagation 41

6.5 UAV state residual propagation 42

6.6 Carrier attitude PSD analysis 45

6.7 Camera residuals 46

6.8 Lidar residuals 46

6.9 GPS residuals 47

6.10 UAV and carrier positions 49

6.11 UAV and carrier velocities 50

6.12 UAV and carrier attitudes 51

6.13 Relative position and velocity estimate 53

6.14 Relative attitude estimate 54

7.1 IMU study - RSS relative position 57

7.2 IMU study - RSS relative velocity and attitude 58

7.3 Camera study - RSS relative position, velocity, and attitude 60

7.4 Lidar study - RSS relative position, velocity, and attitude 61

7.5 GPS study - RSS relative position, velocity, and attitude 63

7.6 Fiducial study - RSS relative position, velocity, and attitude 64

C.1 UAV parameter state errors 86

C.2 Accelerometer and gyroscope biases 87

C.3 Camera and lidar misalignment 88

C.4 Carrier attitude rates 89

C.5 UAV and carrier positions estimate 90

C.6 UAV and carrier velocities estimate 91

C.7 UAV and carrier attitudes estimate 92

C.8 Accelerometer and gyro bias estimate 93

C.9 Camera and lidar misalignment estimate 94

C.10 Carrier attitude rates estimate 95

D.1 IMU study - Relative position and velocity 96

D.2 IMU study - Relative attitude 97

D.3 Camera study - Relative position and velocity 98

D.4 Camera study - Relative attitude 99

D.5 Lidar study - Relative position and velocity 100

D.6 Lidar study - Relative attitude 101

D.7 GPS study - UAV and carrier position, velocity, and attitude 102

D.8 GPS study - Relative position, velocity, and attitude 103

D.9 Fiducial study - UAV and carrier position, velocity, and attitude 104

D.10 Fiducial study - Relative position, velocity, and attitude 105

ACRONYMS

AR	Augmented Reality
EKF	Extended Kalman Filter
GPS	Global Positioning System
IMU	Inertial Measurement Unit
LAMP	Large Amplitude Motions Program
LOS	Line Of Sight
ONR	Office of Naval Research
PSD	Power Spectral Density
RSS	Root Sum Square
SCONE	Systematic Characterization Of the Naval Environment
UAV	Unmanned Aerial Vehicle
VTOL	Vertical Take-Off and Landing

CHAPTER 1

INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are progressively becoming more popular in many fields including military operations, commercial use, and even among hobbyists. Because of the rapidly growing prevalence and technological advancements behind them, UAVs are becoming more intelligent with greater capabilities.

With relatively new breakthroughs in drone[1] design and autopilot software, UAVs are now able to fly fully autonomous missions. These missions can range from simple waypoint following [1] to area surveying [2] and building inspection [3]. However, among the most difficult maneuvers in regard to unmanned flight is the autonomous landing. The landing problem can become even more complicated if the UAV is attempting to land on a moving runway such as a delivery truck or an aircraft carrier.

This paper discusses a system-level analysis of navigation errors in an autonomous UAV carrier-landing scenario. The focus will be on fixed-wing aircraft rather than multi-rotor Vertical Take-off and Landing (VTOL) platforms. The decision to focus on fixed-wing aircraft stems from the aircraft's inability to land without a carefully planned flight trajectory, which leads to more stringent navigation requirements and several opportunities for research. The information and results in this paper provide a thorough analysis of the aircraft/carrier relative estimation system and will enable the development of less expensive landing systems, without compromising safety or accuracy.

1.1 Objectives

With the previous statement as motivation, this work covers and discusses the following three objectives:

[1] The terms "drone" and "UAV" will be used interchangeably in this work.

1. The implementation of an extended Kalman filter that estimates the navigation states of both the UAV and the carrier.

2. The quantification of navigation errors as a function of onboard sensor grade.

3. The analysis of the system's sensitivity to GPS denied environments.

These objectives are accomplished based on a specific system architecture and sensor configuration. Based on the information found in Chapter 2, the following sensors were chosen for the UAV sensor suite: an Inertial Measurement Unit (IMU), a monocular RGB camera, a flash lidar, and a GPS. These sensors were chosen to represent a UAV capable of autonomous landings.

Thus while focusing on the above objectives, the presented research identifies relationships between navigation errors, sensor grade, and sensitivity to position measurements. This will enable designers to select the lowest cost system for a given set of requirements.

1.2 Overview

The remaining chapters of this paper will discuss details pertaining to the project. Chapter 2 reviews the current state of the art in UAV carrier landing scenarios as well as common sensor configurations. This project will focus on a specific sensor configuration which will be further discussed in Chapter 3. This chapter will also cover the materials and design methods. Chapter 4 defines the coordinate frames and state vectors used throughout the paper. Chapter 5 explains the details and implementation of the Kalman filter, while Chapter 6 is dedicated to the verification thereof. Chapter 7 focuses on the results of the paper's objectives. Finally, Chapter 8 summarizes and reiterates the importance of the project.

CHAPTER 2

LITERATURE REVIEW AND BACKGROUND

To better understand the estimation requirements for the landing problem discussed in the introduction, a review of the current state of the art in autonomous UAV landing will be discussed in this chapter.

There are many techniques in the literature for ship-relative localization of incoming UAVs which will be detailed in the following sections. The first section discusses techniques requiring additional infrastructure or sensors attached to the runway. The next section is similar to the first, but these methods require only passive fiducials on the landing zone rather than radios, cameras or other sensors. Section 2.3 covers computer-vision methods for recognizing and tracking the desired landing point. Finally, a section on lidar-aided relative navigation is presented.

As a note, not all of the references and methods discussed here are specific to fixed-wing ship landings. However, the research and designs therein are worth mentioning to introduce common techniques in UAV research that may have impracticalities when applied to fixed-wing aircraft.

2.1 Offboard Sensors

Having additional sensors and hardware on the runway has many benefits for autonomous landing. First off, these systems generally have more computational power available versus a small embedded computer on the UAV. This extra computational power can lead to more accurate and faster recognition algorithms to aid in the estimation process [4]. Secondly, cameras and other tracking systems on the runway are less prone to vibrations and unknown geometries than systems using sensors mounted directly on the UAV [4]. Another benefit that comes from placing sensors on the landing zone rather than the UAV is increased sensor availability. UAV onboard sensors are commonly limited in size, weight,

and power consumption, whereas these limitations are obsolete by placing required sensors offboard. Not only can heavier and larger sensors be used, but the quantity of such sensors is not limited when placed on the runway. This allows arrays of sensors to work together for more accurate measurements [5].

Another approach for gathering state information is by using radios to directly communicate between the ground station and the UAV. With systems such as the one discussed in [6], relative position information can be obtained through pseudo-range and angle of arrival measurements. This direct line of communication reduces estimation errors since the data is being transmitted in real time as the UAV prepares to land.

2.2 Passive Fiducials

Another common system architecture in UAV landings is to place passive fiducials near the landing location. Rather than splitting the computation between the ground station and the UAV, this approach requires all data processing to take place directly on the UAV. This method can be less error prone because it does not rely on direct communication between the ground station and the aircraft, which can also reduce latency issues in the system.

In applications that require only short range position estimates, high-contrast 2D Augmented Reality (AR) markers are able to provide full relative position and orientation measurements from the marker to the UAV [7]. This idea was extended in [8] by embedding multiple markers within a single fiducial and produced much smaller navigation errors than using a single marker.

A downside to AR markers is that from long distances the tags are imperceivable and are thus accurate only at relatively short distances. This limitation makes this method impractical for fixed-wing landings as these aircraft generally require relative pose data at further distances. To overcome these shortcomings, the research carried out in [9] details how leveraging infrared cameras and known visible points on an aircraft carrier can provide ship-relative localization at further distances. Their research determined that a minimum of three passive fiducials were needed to determine the relative position of the UAV.

2.3 Computer-Vision Tracking

In contrast to the above two subsections, computer-vision tracking methods require no additional infrastructure on the landing zone. However, these methods often rely heavily on computer software and camera images that enable simple monocular cameras to recognize and track specific physical features in a video stream. This approach can be employed to guide a UAV towards the desired landing target using only cameras and common computer-vision methods such as the hough transform [10] and RANSAC algorithm [11]. An example of this is found in [12] where an RGB camera and onboard image processing were used to detect and track the runway even if only a small portion was visible. The researchers then implemented a glide-slope controller to guide the UAV to the runway.

Another computer-vision method for gathering UAV state information is referred to as optical flow. This method compares successive camera images to determine the pose of the aircraft relative to objects in the images. As discussed in [13], this can be computationally extensive and, if objects in the images are also moving, can become quite complicated. A common assumption with this method is that relative to the high velocity of the UAV, the objects in the images can be treated as stationary and thus simplify the problem.

To reduce some of the computational complexity, some computer-vision techniques consist of a camera and lidar working together. Introducing the lidar provides direct range information from the UAV to the landing zone. The research conducted in [14] shows that a camera and lidar can be used independently for target detection and tracking. When used together, both sensors produce redundant and complementary state information for a more robust system. This allows for a infrastructure-free ship deck landing.

2.4 Lidar-Aided Relative Navigation

Using lidar in aiding drone landings is not uncommon [15], [16], [17]. Lidar also has applications with spacecraft to survey the surface of celestial objects and determine safe locations for landing [18].

As briefly discussed in Section 2.3, it is common in the literature to use both a camera and lidar for navigation. The research presented in [15] uses a monocular vision camera for

estimating the horizontal position of the landing zone, while the lidar is used primarily for gathering altitude information. Leveraging the height estimate, the descent velocity of the UAV can also be computed based on lidar measurements.

Similarly, [19] combines the range measurements from a lidar with a visual tracking sensor to obtain a simple and low-cost guidance system. This system is capable of determining distance to the carrier and orientation of the ship deck by targeting a narrow-band light source placed along the landing zone. By combining these sensor measurements, an accurate relative estimate of the carrier can be determined.

In [20], a lidar is used for position measurements when a GPS signal is lost. This is accomplished by relying on inertial measurements and by calculating the coordinates of specific points in the environment. These measurements are then used to determine the UAV's location. However, these position measurements tend to drift and errors grow over time without GPS updates.

Given the information in this chapter, a specific system architecture and sensor measurements were chosen and implemented throughout this project. This is further discussed in Chapter 3.

CHAPTER 3

MATERIALS AND METHODS

This chapter focuses on the system setup, the sensor configuration, and the methods used to obtain the project objectives as stated in Chapter 1. The project takes the current state of the art in UAV autonomous carrier landings as addressed in Chapter 2, and determines the navigation sensitivity to sensor errors and environmental factors. This is accomplished by implementing a simulation environment where the estimator can be used to produce accurate and repeatable results. The choice to implement the project in simulation also allows testing many different scenarios in a controlled environment.

In the attempt to avoid confusion, the first section will define notation and conventions used throughout the paper. The sensors used in the navigation system and reasons for the chosen sensor suite will be discussed in section 3.2. The next section, will briefly cover the carrier data used to propagate the carrier's state information throughout the simulation. Finally, the last section of the chapter discusses the EKF and Monte Carlo simulation used to produce the desired results and achieve the paper's objectives.

3.1 Notation and Conventions

In this paper, all vectors are assumed to be column vectors and are denoted with an underline i.e. $\underline{x}$. A dotted variable, i.e. $\dot{x}$, denotes a time derivative. An estimate of a variable is denoted with a hat above the variable i.e. $\hat{x}$, while a measured variable is denoted above with a tilde i.e. $\tilde{x}$.

Direction cosine matrices, also referred to as rotation matrices, are used to represent a rotation between two coordinate frames. These matrices are denoted by an R where the subscript is the starting frame and the superscript is the ending frame. For example, the rotation matrix R_b^n describes a rotation from the b frame to the n frame.

Quaternion Conventions

Rotations will also be described using a unit quaternion denoted by a q. Quaternions follow the same same subscript and superscript notation as a rotation matrix. For example, q_b^n represents a rotation from the b frame to the n frame. It should be noted that each quaternion follows the Hamilton convention with

$$q \triangleq q_0 + q_1\hat{i} + q_2\hat{j} + q_3\hat{k}$$

and when expressing the quaternion in vector form, the scalar is the first element followed by the vector portion.

$$q \triangleq \begin{bmatrix} q_0 \\ \underline{q} \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

A rotation matrix can be created from a quaternion by equation 3.1.1.

$$R(q) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{3.1.1}$$

A quaternion conjugate is denoted as q^* and is defined per equation 3.1.2.

$$q^* \triangleq \begin{bmatrix} q_0 \\ -q_1 \\ -q_2 \\ -q_3 \end{bmatrix} \tag{3.1.2}$$

Quaternion multiplication can be computed as follows:

Given two quaternions

$$p \triangleq \begin{bmatrix} p_0 \\ \underline{p} \end{bmatrix} \quad \text{and} \quad r \triangleq \begin{bmatrix} r_0 \\ \underline{r} \end{bmatrix}$$

$$q = p \otimes r = \begin{bmatrix} p_0 r_0 - \underline{p} \cdot \underline{r} \\ \underline{p} \times \underline{r} + p_0 \underline{r} + r_0 \underline{p} \end{bmatrix} \tag{3.1.3}$$

where $\times$ and $\cdot$ denote a cross product and dot product respectively.

Equations 3.1.1 - 3.1.3 can also be seen in [21].

There can be many definitions of misalignment error in a rotation matrix or quaternion. These can vary depending on which frame is considered misaligned. The definitions for misalignment error used throughout this paper are defined in the following equations which produce equivalent representations of misalignment error.

$$R_b^n \triangleq \left[I - (\underline{\theta}_b^n \times)\right] \hat{R}_b^n \tag{3.1.4}$$

$$q_b^n \triangleq \begin{bmatrix} 1 \\ -\frac{1}{2}\delta\underline{\theta}_b^n \end{bmatrix} \otimes \hat{q}_b^n \tag{3.1.5}$$

3.2 Sensor Suite

Given the background discussed in Chapter 2, a specific sensor suite was chosen for this project that represented common autonomous landing techniques in the literature. The following sensors were chosen to be used and modeled in this project: an Inertial Measurement Unit (IMU), a monocular RGB camera, a flash lidar, and a GPS. All of these sensors are rigidly fixed to the body of the UAV thus requiring no additional offboard sensors on the carrier. However, the carrier is assumed to have passive fiducials in predetermined locations on the ship deck.

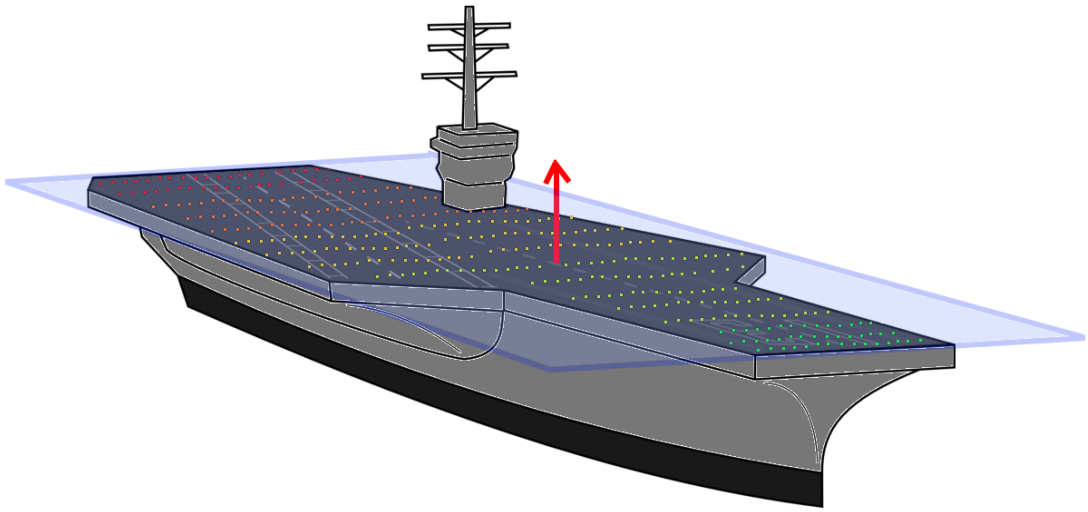

Fig. 3.1: Use of lidar measurement to determine carrier attitude

The IMU is used to propagate the UAV's position, velocity, and attitude states in between measurement updates from the GPS, camera, and lidar. The camera provides line-of-sight (LOS) measurements from the aircraft to the passive fiducials on the carrier. As is typical with most unmanned vehicles, the GPS provides position measurements.

The lidar usage in this project is somewhat unconventional. Instead of using the lidar for range measurements, it is used in estimating the carrier's attitude.[1] This is accomplished in three steps. First, a point cloud of the carrier is created using the flash lidar. Next, a geometric plane is fit to the points that lie on the runway of the ship deck. More information on this process can be found in [23]. Finally, the plane's nominally upward unit vector is extracted containing the current attitude of the carrier. A visual of the lidar measurement can be seen in Figure 3.1.

The decision to use passive fiducials was due to the relative simplicity of the design architecture as opposed to the other methods presented in Chapter 2. This method does not require active infrastructure such as radar or communication links on the ship deck and aside from the additional passive markers on the carrier, keeps the entire landing system onboard the UAV. This means the resulting system is more portable and can be integrated

[1] A similar concept using "beam emitters" can be seen in [22].

into existing carrier systems with minimal modifications. Another reason for the passive fiducials method is that it allows the system to be scaled to smaller platforms that do not possess the required computational power for computer-vision tracking methods. Again, this allows the system to be applied to many different aircraft rather than targeting a specific platform.

3.3 Carrier Model

To accurately model the carrier's behavior in simulation, pose data has been collected from various classes of ships commonly used in the United States Navy. This data was collected and compiled by the Office of Naval Research (ONR) as part of the Systematic Characterization of the Naval Environment (SCONE) project and was created using state of the art nonlinear seakeeping prediction code (LAMP). This work will focus on the Generic Carrier (CVN) ship class which represents typical aircraft carrier hulls. The dataset is used in this work to provide representative motion of the aircraft carrier in the simulation as well as aid in the design of motion models for the extended Kalman filter.

3.4 Estimation and Simulation

Results of the simulation focus on the relative position, velocity and attitude covariances at three seconds prior to the UAV touchdown. This is accomplished using an indirect extended Kalman filter (EKF) and a Monte Carlo simulation. The measurements available to the EKF are camera observations of the passive fiducials on the carrier and lidar-based attitude measurements of the ship deck. The lidar measurements are assumed to be pre-processed to extract the normal vector of the deck plane as discussed in Section 3.2. The sensitivity of these covariances to measurement accuracy and frequency are illustrated as a function of onboard sensor grade.

The successful implementation of the EKF satisfies the first design objective mentioned in the introduction while the analysis of the onboard sensor sensitivity achieves the second objective.

The method for achieving the third objective is to analyze the system's sensitivity to

environmental factors. In the scope of this project, environmental factors will include two cases. The first is the sensitivity of the system to GPS-denied environments where the amount of time without a GPS signal is varied to determine the system's sensitivity and dependence on position measurements. The second case under the environmental factors focuses on the system's navigation errors while varying the number of fiducials on the ship deck. As with the analysis of the onboard sensors, the performance is evaluated based on relative UAV/carrier covariances rather than absolute covariances. The idea of relative covariances is further explained in Section 5.8.

Thus, the research presented in this work enables designers to select system components which minimize cost while maintaining mission-specific navigation error requirements.

CHAPTER 4

STATE DEFINITIONS

This chapter is dedicated to the design model and the variables of interest referred to in this project. The actual value of these variables are grouped into a vector which will be referred to as the truth state vector. The EKF estimates the values of these variables and produces an estimated state vector. The difference between truth and estimated vectors is known as the error state vector.

Section 4.1 explains the relationships between the various coordinate frames in the project. The next section, 4.2, covers the variables in the state vectors. The last section of this chapter discusses the error definitions of the state vector which will later be used in the EKF validation.

4.1 Coordinate Frames

In order to make sense of the variables contained in the state vector, it is necessary to first define the coordinate frames that the variables will be expressed in. Figure 4.1 illustrates the relationships between the six coordinate frames used in this project. All coordinate frames adhere to the right-hand rule with the x-axis denoted by the dashed line in Figure 4.1. The I frame is the inertial frame which is earth-fixed and aligned with the North, East and down directions. The n (navigation) frame is aligned with the I frame, but is fixed at the center of mass of the UAV. The b (body) frame is coincident with the n frame at the UAV's center of mass, but the x-axis points out the nose of the UAV and rotates with the body of the aircraft. The c (carrier) frame behaves like the b frame, but is attached to the carrier instead of the UAV, with the x-axis pointing out the front of the carrier. The final two frames are attached to the camera (k frame) and the lidar (l frame) which are both rigidly fixed to the UAV. In this paper, the origin of these two frames are assumed to be coincident with the origin of the b frame. Thus, the $\underline{k}^b$ and the $\underline{l}^b$ vectors are

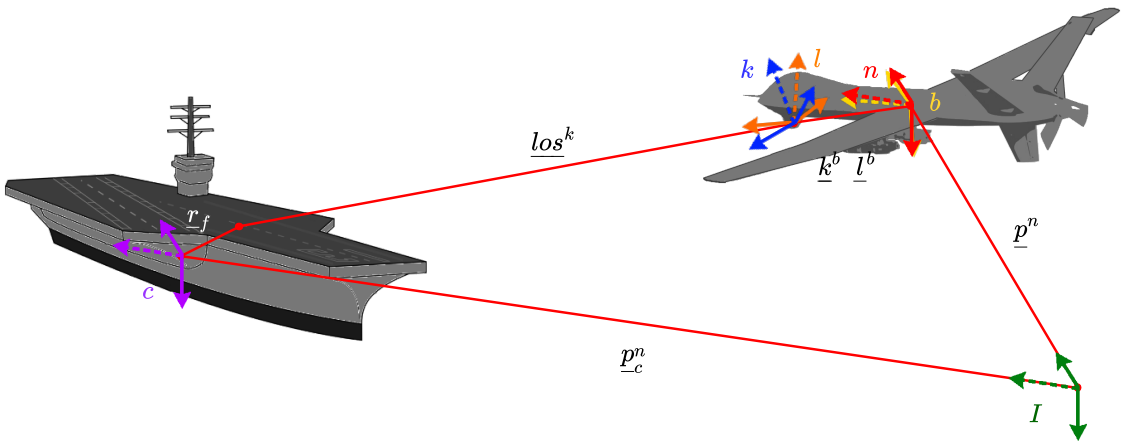

Fig. 4.1: Coordinate frame diagram

zero which means that these sensors are assumed to be a the center of mass of the UAV. Although it it not realistic to have both the camera and lidar fixed at the center of mass of the UAV, eliminating these lever arms simplifies the analysis without significantly affecting the results of the study. This is largely due to the magnitude of the position vectors that dominate the geometry in the landing scenario.

4.2 State Vectors

The truth, estimated, and error state vectors are denoted as $\underline{x}$, $\underline{\hat{x}}$, and $\delta\underline{x}$ respectively. Each vector is an 14×1 column vector, but will be represented as a transposed row vector as shown below. Table 4.1 details each of the variables contained in the state vector.

$$\underline{x} = \begin{bmatrix} \underline{p}^n & \underline{v}^n & q_b^n & \underline{b}_a & \underline{b}_g & q_k^b & q_l^b & \underline{p}_c^n & \underline{v}_c^n & q_c^n & \omega_{cx/n}^c & \alpha_{cx/n}^c & \omega_{cy/n}^c & \omega_{cz/n}^c \end{bmatrix}^T$$

$$\underline{\hat{x}} = \begin{bmatrix} \underline{\hat{p}}^n & \underline{\hat{v}}^n & \hat{q}_b^n & \underline{\hat{b}}_a & \underline{\hat{b}}_g & \hat{q}_k^b & \hat{q}_l^b & \underline{\hat{p}}_c^n & \underline{\hat{v}}_c^n & \hat{q}_c^n & \hat{\omega}_{cx/n}^c & \hat{\alpha}_{cx/n}^c & \hat{\omega}_{cy/n}^c & \hat{\omega}_{cz/n}^c \end{bmatrix}^T$$

$$\delta\underline{x} = \begin{bmatrix} \delta\underline{p}^n & \delta\underline{v}^n & \delta\underline{\theta}_b^n & \delta\underline{b}_a & \delta\underline{b}_g & \delta\underline{\theta}_k^b & \delta\underline{\theta}_l^b & \delta\underline{p}_c^n & \delta\underline{v}_c^n & \delta\underline{\theta}_c^n & \delta\omega_{cx/n}^c & \delta\alpha_{cx/n}^c & \delta\omega_{cy/n}^c & \delta\omega_{cz/n}^c \end{bmatrix}^T$$

Table 4.1: State variable descriptions

Variable Name	**Description**
$\underline{p}^n$	Position of the UAV as expressed in the navigation frame
$\underline{v}^n$	Velocity of the UAV as expressed in the navigation frame
q_b^n	Quaternion from the body frame to the navigation frame
$\underline{b}_a$	Accelerometer bias in the body frame
$\underline{b}_g$	Gyroscope bias in the body frame
q_k^b	Quaternion from the camera frame to the body frame
q_l^b	Quaternion from the lidar frame to the body frame
$\underline{p}_c^n$	Position of the carrier as expressed in the navigation frame
$\underline{v}_c^n$	Velocity of the carrier as expressed in the navigation frame
q_c^n	Quaternion from the carrier frame to the navigation frame
$\omega_{cx/n}^c$	Angular velocity about the x-axis of the carrier with respect to the navigation frame as expressed in the carrier frame
$\alpha_{cx/n}^c$	Internal state used in the calculation of $\omega_{cx/n}^c$
$\omega_{cy/n}^c$	Angular velocity about the y-axis of the carrier with respect to the navigation frame as expressed in the carrier frame
$\omega_{cz/n}^c$	Angular velocity about the z-axis of the carrier with respect to the navigation frame as expressed in the carrier frame

Each variable denoted with an underline within the state vectors are 3×1 vectors themselves. For example, $\underline{p}^n$ denotes the position of the UAV expressed in the navigation frame. This vector contains the x, y, and z positions of the UAV. For simplicity, the state vectors can be consolidated and represented as

$$\underline{x} = \begin{bmatrix} \underline{x}_u & \underline{x}_p & \underline{x}_c \end{bmatrix}^T \tag{4.2.1}$$

where the elements $\underline{x}_u$, $\underline{x}_p$, and $\underline{x}_c$ are defined as

$$\underline{x}_u = \begin{bmatrix} \underline{p}^n & \underline{v}^n & q_b^n \end{bmatrix}^T \tag{4.2.2}$$

$$\underline{x}_p = \begin{bmatrix} \underline{b}_a & \underline{b}_g & q_k^b & q_l^b \end{bmatrix}^T \tag{4.2.3}$$

$$\underline{x}_c = \begin{bmatrix} \underline{p}_c^n & \underline{v}_c^n & q_c^n & \omega_{cx/n}^c & \alpha_{cx/n}^c & \omega_{cy/n}^c & \omega_{cz/n}^c \end{bmatrix}^T \tag{4.2.4}$$

4.3 Error Definitions

The matrix equations in this section show the relationships between the truth, estimated, and error state vectors. Since the EKF can only produce an estimate of the state, the value of the truth vector, $\underline{x}$, is a small perturbation from the estimated value $\hat{\underline{x}}$. This small perturbation is referred to as the error state or $\delta\underline{x}$. In many cases, this relationship is simply additive as in the case for the position state, $\underline{p}^n = \hat{\underline{p}}^n + \delta\underline{p}^n$. The equation for the truth quaternion requires a quaternion multiplication of the estimated quaternion and a δq term which is defined in equation 4.3.1.

$$\delta q_x^y \triangleq \begin{bmatrix} 1 \\ -\frac{1}{2}\delta\underline{\theta}_x^y \end{bmatrix} \tag{4.3.1}$$

where δq_x^y is a small perturbation in the quaternion that represents a rotation from the x frame to the y frame. The variables x and y are used as placeholders to represent each of the δq terms found below. The full mapping from estimated state and perturbations to

truth state is shown in equation 4.3.2.

$$\underline{x} = \begin{bmatrix} \underline{\hat{p}}^n + \delta\underline{p}^n \\ \underline{\hat{v}}^n + \delta\underline{v}^n \\ \delta q_b^n \otimes \hat{q}_b^n \\ \underline{\hat{b}}_a + \delta\underline{b}_a \\ \underline{\hat{b}}_g + \delta\underline{b}_g \\ \delta q_k^b \otimes \hat{q}_k^b \\ \delta q_l^b \otimes \hat{q}_l^b \\ \underline{\hat{p}}_c^n + \delta\underline{p}_c^n \\ \underline{\hat{v}}_c^n + \delta\underline{v}_c^n \\ \delta q_c^n \otimes \hat{q}_c^n \\ \hat{\omega}_{cx/n}^c + \delta\omega_{cx/n}^c \\ \hat{\alpha}_{cx/n}^c + \delta\alpha_{cx/n}^c \\ \hat{\omega}_{cy/n}^c + \delta\omega_{cy/n}^c \\ \hat{\omega}_{cz/n}^c + \delta\omega_{cz/n}^c \end{bmatrix} \tag{4.3.2}$$

Where $\otimes$ denotes quaternion multiplication.

The estimated state can be defined in terms of the truth state and a small perturbation from that state. Thus the mapping for $\underline{\hat{x}}$ is defined as follows where * denotes the quaternion conjugate.

$$\hat{\underline{x}} = \begin{bmatrix} \underline{p}^n - \delta\underline{p}^n \\ \underline{v}^n - \delta\underline{v}^n \\ (\delta q_b^n)^* \otimes q_b^n \\ \underline{b}_a - \delta\underline{b}_a \\ \underline{b}_g - \delta\underline{b}_g \\ (\delta q_k^b)^* \otimes q_k^b \\ (\delta q_l^b)^* \otimes q_l^b \\ \underline{p}_c^n - \delta\underline{p}_c^n \\ \underline{v}_c^n - \delta\underline{v}_c^n \\ (\delta q_c^n)^* \otimes q_c^n \\ \omega_{cx/n}^c - \delta\omega_{cx/n}^c \\ \alpha_{cx/n}^c - \delta\alpha_{cx/n}^c \\ \omega_{cy/n}^c - \delta\omega_{cy/n}^c \\ \omega_{cz/n}^c - \delta\omega_{cz/n}^c \end{bmatrix} \tag{4.3.3}$$

The final variation of these matrix mappings is a function of truth and estimated state and produces the perturbation. In most cases, this is accomplished by subtracting the estimated state from the truth vector, however, another quaternion multiplication is required for the quaternion elements in the vector. Equation 4.3.4 shows this mapping.

$$\delta\underline{x} = \begin{bmatrix} \underline{p}^n - \hat{\underline{p}}^n \\ \underline{v}^n - \hat{\underline{v}}^n \\ \delta\underline{\theta}_b^n \\ \underline{b}_a - \hat{\underline{b}}_a \\ \underline{b}_g - \hat{\underline{b}}_g \\ \delta\underline{\theta}_k^b \\ \delta\underline{\theta}_l^b \\ \underline{p}_c^n - \hat{\underline{p}}_c^n \\ \underline{v}_c^n - \hat{\underline{v}}_c^n \\ \delta\underline{\theta}_c^n \\ \omega_{cx/n}^c - \hat{\omega}_{cx/n}^c \\ \alpha_{cx/n}^c - \hat{\alpha}_{cx/n}^c \\ \omega_{cy/n}^c - \hat{\omega}_{cy/n}^c \\ \omega_{cz/n}^c - \hat{\omega}_{cz/n}^c \end{bmatrix} \tag{4.3.4}$$

where $\delta\underline{\theta}_x^y \triangleq -2 \times [q_x^y \otimes (\hat{q}_x^y)^*](2:4)$

The $(2:4)$ notation denotes the extraction of the 2 - 4 elements of the resulting 4×1 column vector. The variables x and y are used as placeholders to represent each of the $\delta\underline{\theta}$ terms in equation 4.3.4.

The following chapter covers how the state vector evolves over time governed by the state propagation equations.

CHAPTER 5

NAVIGATION AND STATE ESTIMATION

This chapter is dedicated to the development and implementation of the extended Kalman filter. Sections 5.1 and 5.2, present the nonlinear equations governing how the state vectors propagate through time. Section 5.3 includes the nonlinear models representing the camera, lidar, and GPS measurements used in the EKF. Sections 5.4 and 5.5, take the nonlinear design and measurement models and linearizes them to produce models that are easily integrated into the EKF. This linearization process also provides a way to accurately propagate the state covariances. Section 5.6 documents the covariance propagation, while the following Section, 5.7, presents the Kalman update equations and briefly discusses the recursive Kalman filter loop. The final section in this chapter documents the relative covariance transforms.

5.1 Nonlinear Truth State Propagation

Each of the states within the state vector require a differential equation to mathematically describe how the state evolves over time. These equations are explained below in the order they appear in the state vector.

The derivative of the UAV position is the UAV velocity.

$$\dot{\underline{p}}^n = \underline{v}^n \tag{5.1.1}$$

The derivative of the UAV's velocity is the specific force on the UAV, denoted as $\underline{\nu}^b$, plus gravity. However, since the acceleration needs to be expressed in the navigation frame, this value is then multiplied by the rotation matrix R_b^n.

$$\dot{\underline{v}}^n = R_b^n \underline{\nu}^b + \underline{g}^n \tag{5.1.2}$$

The quaternion dynamics relating the body frame and the navigation frame requires a quaternion multiplication of the original quaternion scaled by 1/2 and an augmented quaternion of zero and the angular velocity of the body frame. This relationship is shown below.

$$\dot{q}_b^n = \frac{1}{2} q_b^n \otimes \begin{bmatrix} 0 \\ \underline{\omega}^b \end{bmatrix} \tag{5.1.3}$$

The next two equations are first order Markov processes for the accelerometer and the gyroscope biases. The derivative of these biases are easily calculated by scaling their value by their respective time constants with the addition of a noise term, $\underline{n}$.

$$\dot{\underline{b}}_a = -\frac{1}{\tau_a} \underline{b}_a + \underline{n}_a \tag{5.1.4}$$

$$\dot{\underline{b}}_g = -\frac{1}{\tau_g} \underline{b}_g + \underline{n}_g \tag{5.1.5}$$

The following two quaternions represent the rotation from camera frame to the body frame as well as from the lidar frame to the body frame. Since these two frames are rigidly attached to the UAV body frame, they do not evolve over time.

$$\dot{q}_k^b = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{5.1.6}$$

$$\dot{q}_l^b = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{5.1.7}$$

Equations 5.1.1 - 5.1.7 are once again consolidated into matrix form

$$\dot{\underline{x}}_u = \begin{bmatrix} \underline{v}^n \\ R_b^n \underline{\nu}^b + \underline{g}^n \\ \frac{1}{2} q_b^n \otimes \begin{bmatrix} 0 \\ \underline{\omega}^b \end{bmatrix} \end{bmatrix} \tag{5.1.8}$$

$$\dot{\underline{x}}_p = \begin{bmatrix} -\frac{1}{\tau_a}\underline{b}_a + \underline{n}_a \\ -\frac{1}{\tau_g}\underline{b}_g + \underline{n}_g \\ \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T \\ \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T \end{bmatrix} \tag{5.1.9}$$

The derivative of the variables contained in $\underline{x}_c$ as defined in equation 4.2.4, correspond to the carrier states. For the truth vector, these values are read from a file containing the data recorded on an aircraft carrier as discussed in Section 3.3. This data was collected and compiled by the Office of Naval Research (ONR) and made public to academic researchers. Thus, the recorded carrier data was used as the truth rather than propagating these values in simulation.

Similar UAV dynamics can be found in [24]. However, the velocities used in this paper are expressed in the navigation frame which yield slightly different equations. The equations used in the navigation filter to estimate these values are described in the next section.

5.2 Nonlinear Navigation Differential Equations

The equations governing the UAV states in this section are almost identical to the truth state propagation equations without the noise terms. The estimated state propagation equations for these states are shown below:

$$\dot{\hat{\underline{x}}}_u = \begin{bmatrix} \hat{\underline{v}}^n \\ \hat{R}_b^n(\tilde{\underline{\nu}}^b - \hat{\underline{b}}_a) + \underline{g}^n \\ \frac{1}{2}\hat{q}_b^n \otimes \begin{bmatrix} 0 \\ \tilde{\underline{\omega}}^b - \hat{\underline{b}}_g \end{bmatrix} \end{bmatrix} \tag{5.2.1}$$

$$\dot{\hat{\underline{x}}}_p = \begin{bmatrix} & -\frac{1}{\tau_a}\underline{b}_a & \\ & -\frac{1}{\tau_g}\underline{b}_g & \\ \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T & & \\ \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T & & \end{bmatrix} \tag{5.2.2}$$

As seen from above, the only difference between these equations and their truth counterparts are equations for the velocity and attitude. These two equations now subtract off their corresponding biases from their measurements $\tilde{\underline{\nu}}^b$ and $\tilde{\underline{\omega}}^b$ which are defined below in 5.2.3 and 5.2.4 from the accelerometer and gyroscope respectively.

$$\tilde{\underline{\nu}}^b \triangleq \underline{\nu}^b + \underline{b}_a + \underline{\eta}_\nu \tag{5.2.3}$$

$$\tilde{\underline{\omega}}^b \triangleq \underline{\omega}^b + \underline{b}_g + \underline{\eta}_\omega \tag{5.2.4}$$

where $\underline{\eta}_\nu$ and $\underline{\eta}_\omega$ are the noise corruptions in the sensors themselves.

The estimated state equations for position and attitude of the carrier are identical to the UAV truth state equations 5.1.1 and 5.1.3.

$$\dot{\hat{\underline{p}}}_c^n = \hat{\underline{v}}_c^n \tag{5.2.5}$$

$$\dot{\hat{q}}_c^n = \frac{1}{2}\hat{q}_c^n \otimes \begin{bmatrix} 0 \\ \hat{\underline{\omega}}^c \end{bmatrix} \tag{5.2.6}$$

The velocity of the carrier is slightly different. Since the velocity of the carrier is assumed to be nominally constant throughout the simulation, the propagation of this state is modeled as a random walk as shown below.

$$\dot{\hat{\underline{v}}}_c^n = \underline{n}_{vc} \tag{5.2.7}$$

The remaining states are related to the angular velocity of the carrier where $\omega^c_{cx/n}$, $\omega^c_{cy/n}$, and $\omega^c_{cz/n}$ are the velocities about their respective axes. The roll of the carrier about the x-axis has a resonant peak at a specific frequency which behaves like a second order Markov process. Because of the second order system, $\alpha^c_{cx/n}$ is used as an internal state for the dynamics of the carrier about this axis. The natural frequency, ω_n, and the damping ratio, ζ, are tuning parameters used to accurately describe the roll of the carrier. The pitch and yaw of the carrier are modeled using a first order Markov as was done with the accelerometer and gyroscope biases. The following four equations describe these relationships.

$$\dot{\hat{\omega}}^c_{cx/n} = \hat{\alpha}^c_{cx/n} \tag{5.2.8}$$

$$\dot{\hat{\alpha}}^c_{cx/n} = -\omega_n^2 \hat{\omega}^c_{cx/n} - 2\zeta\omega_n \hat{\alpha}^c_{cx/n} + n_{\omega cx}\omega_n^2 \tag{5.2.9}$$

$$\dot{\hat{\omega}}^c_{cy/n} = -\frac{1}{\tau_{\omega cy}} \hat{\omega}^c_{cy/n} + n_{\omega cy} \tag{5.2.10}$$

$$\dot{\hat{\omega}}^c_{cz/n} = -\frac{1}{\tau_{\omega cz}} \hat{\omega}^c_{cz/n} + n_{\omega cz} \tag{5.2.11}$$

Thus the equation for $\dot{\underline{\hat{x}}}_c$ is as follows

$$\dot{\underline{\hat{x}}}_c = \begin{bmatrix} \underline{\hat{v}}^n_c \\ \underline{n}_{vc} \\ \frac{1}{2}\hat{q}^n_c \otimes \begin{bmatrix} 0 \\ \underline{\hat{\omega}}^c \end{bmatrix} \\ \hat{\alpha}^c_{cx/n} \\ -\omega_n^2 \hat{\omega}^c_{cx/n} - 2\zeta\omega_n \hat{\alpha}^c_{cx/n} + n_{\omega cx}\omega_n^2 \\ -\frac{1}{\tau_{\omega cy}} \hat{\omega}^c_{cy/n} + n_{\omega cy} \\ -\frac{1}{\tau_{\omega cz}} \hat{\omega}^c_{cz/n} + n_{\omega cz} \end{bmatrix} \tag{5.2.12}$$

Section 6.3 contains more information on the process of determining the tuned carrier parameters.

5.3 Nonlinear Measurement Models

The UAV has three sensors onboard that require a model to represent their measurements: the camera, the lidar and the GPS. The models for each of these sensors will be discussed below.

Camera

The camera provides a line-of-sight (LOS) measurement from the UAV to the carrier landing zone. The camera model is shown below in equation 5.3.1

$$\tilde{\underline{z}}_k = \begin{bmatrix} l_x/l_z \\ l_y/l_z \end{bmatrix} + \underline{\eta}_k \tag{5.3.1}$$

where l is the LOS vector from the UAV camera frame to the fiducial in the carrier frame and $\underline{n}_k$ is the camera noise. $\underline{l}$ must be expressed in the camera frame and can be defined as follows:

$$\underline{l}^k \triangleq \underline{p}_c^k + \underline{r}_f^k - \underline{p}^k$$

Adding a few rotation matrices is required to express the above equation in terms of states in the state vector. The resulting equation becomes:

$$\underline{l}^k \triangleq R_b^k R_n^b \underline{p}_c^n + R_b^k R_n^b R_c^n \underline{r}_f^c - R_b^k R_n^b \underline{p}^n \tag{5.3.2}$$

Which can equivalently be expressed as:

$$\underline{l}^k \triangleq R_b^k R_n^b \left(\underline{p}_c^n + R_c^n \underline{r}_f^c - \underline{p}^n \right) \tag{5.3.3}$$

Lidar

Similar to the camera, the lidar provides a range measurement from the UAV lidar frame to the carrier frame. The lidar produces a point cloud of the ship deck which is then converted to a unit normal vector, $\underline{u}$. This normal vector is the measurement extracted from the lidar and is defined in 5.3.4.

$$\underline{u}^c \triangleq \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \tag{5.3.4}$$

Since this vector is defined in the carrier frame, its values are constant and always points in the negative z direction of the carrier frame. As was done with the camera measurement, the normal vector is then rotated into the lidar frame to produce the following lidar measurement model:

$$\tilde{\underline{z}}_l = L_m R_b^l R_n^b R_c^n \underline{u}^c + \underline{\eta}_l \tag{5.3.5}$$

where L_m is a matrix that maps the 3 dimensional unit vector to the x and y components of the lidar frame

$$L_m \triangleq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{5.3.6}$$

GPS

The final model describes the GPS measurements of the UAV. As this information is already in the state vector, the model is simply the p^n vector with additional noise.

$$\tilde{\underline{z}}_{gps} = \underline{p}^n + \underline{\eta}_{gps} \tag{5.3.7}$$

Residuals

Now that each of the measurement models have been formed, the estimate of these models are defined with the same equations except with values from the $\hat{\underline{x}}$ vector instead of $\underline{x}$ and without the noise terms.

$$\hat{\tilde{\underline{z}}}_k = \begin{bmatrix} \hat{l}_x/\hat{l}_z \\ \hat{l}_y/\hat{l}_z \end{bmatrix} \tag{5.3.8}$$

$$\hat{\tilde{\underline{z}}}_l = \hat{R}_b^l \hat{R}_n^b \hat{R}_c^n \hat{\underline{u}}^c \tag{5.3.9}$$

$$\hat{\underline{\tilde{z}}}_{gps} = \hat{\underline{p}}^n \tag{5.3.10}$$

With all the measurements and measurement estimates defined, the residuals are computed as follows:

$$\underline{Res}_k \triangleq \underline{\tilde{z}}_k - \hat{\underline{\tilde{z}}}_k \tag{5.3.11}$$

$$\underline{Res}_l \triangleq \underline{\tilde{z}}_l - \hat{\underline{\tilde{z}}}_l \tag{5.3.12}$$

$$\underline{Res}_{gps} \triangleq \underline{\tilde{z}}_{gps} - \hat{\underline{\tilde{z}}}_{gps} \tag{5.3.13}$$

5.4 Linearized Design Model

Now that the nonlinear models have been defined, they must be linearized about the current state estimate to develop linear perturbation models of the state dynamics and measurements. This section covers the linearization of the design and discussed in Section 5.2. This is accomplished by expanding the nonlinear equations about the current estimated states. The linear system is produced by subtracting the nominal states and canceling second-order terms which results in a first-order linearization of the system dynamics. Appendix A.4 discusses the validity of this linear approximation.

The $\dot{\underline{x}}$ equation can be defined by taking the derivative of each of the states in equation 4.3.2. Recalling the definitions for $\tilde{\underline{\nu}}^b$ and $\tilde{\underline{\omega}}^b$ from equations 5.2.3 and 5.2.4, and substituting in equations 5.1.1 - 5.1.7 for the UAV states and 5.2.5 - 5.2.11 for the carrier states yields the following equations.

$$\dot{\hat{\underline{p}}} + \delta\dot{\underline{p}}^n = \hat{\underline{v}}^n + \delta\underline{v}^n \tag{5.4.1}$$

$$\dot{\hat{\underline{v}}}^n + \delta\dot{\underline{v}}^n = \left[I - (\underline{\theta}_b^n\times)\right]\hat{R}_b^n\left(\tilde{\underline{\nu}}^b - \hat{\underline{b}}_a - \delta\underline{b}_a - \underline{\eta}_\nu\right) + \underline{g}^n \tag{5.4.2}$$

$$\frac{d}{dt}\left(\begin{bmatrix}1\\ -\frac{1}{2}\delta\underline{\theta}_b^n\end{bmatrix}\otimes\hat{q}_b^n\right) = \frac{1}{2}\begin{bmatrix}1\\ -\frac{1}{2}\delta\underline{\theta}_b^n\end{bmatrix}\otimes\hat{q}_b^n\otimes\begin{bmatrix}0\\ \tilde{\underline{\omega}}^b - \hat{\underline{b}}_g - \delta\underline{b}_g - \underline{\eta}_\omega\end{bmatrix} \tag{5.4.3}$$

$$\dot{\hat{\underline{b}}}_a + \delta\dot{\underline{b}}_a = -\frac{1}{\tau_a}\left(\hat{\underline{b}}_a + \delta\underline{b}_a\right) + \underline{n}_a \tag{5.4.4}$$

$$\dot{\hat{\underline{b}}}_g + \delta\dot{\underline{b}}_g = -\frac{1}{\tau_g}\left(\hat{\underline{b}}_g + \delta\underline{b}_g\right) + \underline{n}_g \tag{5.4.5}$$

$$\frac{d}{dt}\left(\begin{bmatrix}1\\ -\frac{1}{2}\delta\underline{\theta}_k^b\end{bmatrix} \otimes \hat{q}_k^b\right) = \frac{1}{2}\begin{bmatrix}1\\ -\frac{1}{2}\delta\underline{\theta}_k^b\end{bmatrix} \otimes \hat{q}_k^b \otimes \begin{bmatrix}0\\0\\0\\0\end{bmatrix} \tag{5.4.6}$$

$$\frac{d}{dt}\left(\begin{bmatrix}1\\ -\frac{1}{2}\delta\underline{\theta}_l^b\end{bmatrix} \otimes \hat{q}_l^b\right) = \frac{1}{2}\begin{bmatrix}1\\ -\frac{1}{2}\delta\underline{\theta}_l^b\end{bmatrix} \otimes \hat{q}_l^b \otimes \begin{bmatrix}0\\0\\0\\0\end{bmatrix} \tag{5.4.7}$$

$$\dot{\hat{\underline{p}}}_c + \delta\dot{\underline{p}}_c^n = \hat{\underline{v}}_c^n + \delta\underline{v}_c^n \tag{5.4.8}$$

$$\dot{\hat{\underline{v}}}_c^n + \delta\dot{\underline{v}}_c^n = \left[I - (\underline{\theta}_c^n\times)\right]\hat{R}_b^n\left(-\underline{n}_{vc}\right) \tag{5.4.9}$$

$$\frac{d}{dt}\left(\begin{bmatrix}1\\ -\frac{1}{2}\delta\underline{\theta}_c^n\end{bmatrix} \otimes \hat{q}_c^n\right) = \frac{1}{2}\begin{bmatrix}1\\ -\frac{1}{2}\delta\underline{\theta}_c^n\end{bmatrix} \otimes \hat{q}_c^n \otimes \begin{bmatrix}0\\ \underline{\omega}_{c/n}^c - \delta\underline{\omega}_{c/n}^c - \underline{n}_{\omega c}\end{bmatrix} \tag{5.4.10}$$

$$\dot{\hat{\omega}}_{cx/n}^c + \delta\dot{\omega}_{cx/n}^c = \hat{\alpha}_{cx/n}^c + \delta\alpha_{cx/n}^c \tag{5.4.11}$$

$$\dot{\hat{\alpha}}_{cx/n}^c + \delta\dot{\alpha}_{cx/n}^c = -\omega_n^2\left(\hat{\omega}_{cx/n}^c + \delta\omega_{cx/n}^c\right) - 2\zeta\omega_n\left(\hat{\alpha}_{cx/n}^c + \delta\hat{\alpha}_{cx/n}^c\right) \tag{5.4.12}$$

$$\dot{\hat{\omega}}_{cy/n}^c + \delta\dot{\omega}_{cy/n}^c = -\frac{1}{\tau_{\omega cy}}\left(\hat{\omega}_{cy/n}^c + \delta\omega_{cy/n}^c\right) + n_{\omega cy} \tag{5.4.13}$$

$$\dot{\hat{\omega}}_{cz/n}^c + \delta\dot{\omega}_{cz/n}^c = -\frac{1}{\tau_{\omega cz}}\left(\hat{\omega}_{cz/n}^c + \delta\omega_{cz/n}^c\right) + n_{\omega cz} \tag{5.4.14}$$

For each of the equations listed above except the quaternions and velocities, the corresponding linear equation is produced by subtracting the nominal states as expressed in equation 5.2.12. This is known as a perturbation linearization. The velocity and quaternion

equations require a more involved derivation and can be found in appendices A.1 and A.2 respectively. The final linear equations are shown below:

$$\delta\dot{\underline{p}}^n = \delta\underline{v}^n \tag{5.4.15}$$

$$\delta\dot{\underline{v}}^n = \left[\hat{R}_b^n\left(\tilde{\underline{\nu}}^b - \hat{\underline{b}}_a\right)\right]\times\delta\underline{\theta}_b^n - \hat{R}_b^n\delta\underline{b}_a - \hat{R}_b^n\underline{\eta}_\nu \tag{5.4.16}$$

$$\delta\dot{\underline{\theta}}_b^n = \hat{R}_b^n\delta\underline{b}_g + \hat{R}_b^n\underline{\eta}_\omega \tag{5.4.17}$$

$$\delta\dot{\underline{b}}_a = -\frac{1}{\tau_a}\delta\underline{b}_a + \underline{n}_a \tag{5.4.18}$$

$$\delta\dot{\underline{b}}_g = -\frac{1}{\tau_g}\delta\underline{b}_g + \underline{n}_g \tag{5.4.19}$$

$$\delta\dot{\underline{\theta}}_k^b = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^T \tag{5.4.20}$$

$$\delta\dot{\underline{\theta}}_l^b = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^T \tag{5.4.21}$$

$$\delta\dot{\underline{p}}_c^n = \delta\underline{v}_c^n \tag{5.4.22}$$

$$\delta\dot{\underline{v}}_c^n = \underline{n}_{vc} \tag{5.4.23}$$

$$\delta\dot{\underline{\theta}}_c^n = -\hat{R}_c^n\delta\underline{\omega}_c \tag{5.4.24}$$

$$\delta\dot{\omega}_{cx/n}^c = \delta\alpha_{cx/n}^c \tag{5.4.25}$$

$$\delta\dot{\alpha}_{cx/n}^c = -\omega_n^2\delta\omega_{cx/n}^c - 2\zeta\omega_n\delta\alpha_{cx/n}^c + n_{\omega cx}\omega_n^2 \tag{5.4.26}$$

$$\delta\dot{\omega}_{cy/n}^c = -\frac{1}{\tau_{\omega cy}}\delta\omega_{cy/n}^c + n_{\omega cy} \tag{5.4.27}$$

$$\delta\dot{\omega}_{cz/n}^c = -\frac{1}{\tau_{\omega cz}}\delta\omega_{cz/n}^c + n_{\omega cz} \tag{5.4.28}$$

Now that the system is linear, the above equations can be represented as a matrix, F, in the following form.

$$\delta\dot{\underline{x}} = F\left(\hat{\underline{x}}\right)\delta\underline{x} + B\underline{w} \tag{5.4.29}$$

where F is a 34×34 element matrix defined in equation 5.4.30 as a 2×2 block-diagonal matrix.

$$F = \begin{bmatrix} F_{u,p} & 0_{21\times 10} \\ 0_{13\times 24} & F_c \end{bmatrix} \tag{5.4.30}$$

where the $F_{u,p}$ and F_c matrices are defined below:

$$F_{u,p} = \begin{bmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times9} \\ 0_{3\times3} & 0_{3\times3} & C_{\theta_b^n} & -\hat{R}_b^n & 0_{3\times3} & 0_{3\times9} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \hat{R}_b^n & 0_{3\times9} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -\frac{1}{\tau_a} I_{3\times3} & 0_{3\times3} & 0_{3\times9} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -\frac{1}{\tau_g} I_{3\times3} & 0_{3\times9} \\ 0_{6\times3} & 0_{6\times3} & 0_{6\times3} & 0_{6\times3} & 0_{6\times3} & 0_{6\times9} \end{bmatrix} \tag{5.4.31}$$

$$F_c = \begin{bmatrix} I_{3\times3} & 0_{3\times3} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} \\ 0_{3\times3} & 0_{3\times3} & R1 & 0_{3\times1} & R2 & R3 \\ 0_{1\times3} & 0_{1\times3} & 0 & 1 & 0 & 0 \\ 0_{1\times3} & 0_{1\times3} & -\omega_n^2 & -2\zeta\omega_n & 0 & 0 \\ 0_{1\times3} & 0_{1\times3} & 0 & 0 & -\frac{1}{\tau_{\omega y}} & 0 \\ 0_{1\times3} & 0_{1\times3} & 0 & 0 & 0 & -\frac{1}{\tau_{\omega z}} \end{bmatrix} \tag{5.4.32}$$

where $C_{\theta_b^n}$ is the cross product matrix defined as $C_{\theta_b^n} \triangleq \begin{bmatrix} 0 & -c_3 & c_2 \\ c_3 & 0 & -c_1 \\ -c_2 & c_1 & 0 \end{bmatrix}$ and $\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \triangleq \hat{R}_b^n \left(\underline{\tilde{\nu}}^b - \underline{\hat{b}}_a \right)$

The elements $R_{c_1}^n$, $R_{c_2}^n$, and $R_{c_3}^n$ are the first, second, and third columns of the rotation

matrix from the carrier to the navigation frame, R_c^n. For reference, the complete F matrix is shown in Appendix A.3

The noise vector, $\underline{w}$, is an augmentation of all the process noise vectors, while B is the process noise coupling matrix that maps the noise vector to the error state as is defined as

$$B = \begin{bmatrix} B_{u,p} & 0_{24\times 6} \\ 0_{10\times 12} & B_c \end{bmatrix} \tag{5.4.33}$$

$$B_{u,p} = \begin{bmatrix} 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} \\ -\hat{R}_b^n & 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & \hat{R}_b^n & 0_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} & I_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} & I_{3\times 3} \\ 0_{6\times 3} & 0_{6\times 3} & 0_{6\times 3} & 0_{6\times 3} \end{bmatrix} \tag{5.4.34}$$

$$B_c = \begin{bmatrix} I_{3\times 3} & 0_{3\times 1} & 0_{3\times 1} & 0_{3\times 1} \\ 0_{3\times 3} & 0_{3\times 1} & 0_{3\times 1} & 0_{3\times 1} \\ 0_{1\times 3} & 0 & 0 & 0 \\ 0_{1\times 3} & \omega_{n^2} & 0 & 0 \\ 0_{1\times 3} & 0 & 1 & 0 \\ 0_{1\times 3} & 0 & 0 & 1 \end{bmatrix} \tag{5.4.35}$$

$$\underline{w} = \begin{bmatrix} \underline{\eta}_{\nu} \\ \underline{\eta}_{\omega} \\ \underline{n}_{a} \\ \underline{n}_{g} \\ \underline{n}_{vc} \\ n_{\omega cx} \\ n_{\omega cy} \\ n_{\omega cz} \end{bmatrix} \tag{5.4.36}$$

5.5 Linearized Measurement Model

The same procedure was followed for linearizing the measurement models as described in Section 5.3. The discrete measurements were linearized to produce a matrix equation of the form

$$\delta \underline{z} = H\left(\hat{\underline{x}}\right) \delta \underline{x} + G \underline{\nu} \tag{5.5.1}$$

where H is an 7×34 matrix that maps the perturbation state $\delta \underline{x}$ to the linear measurement perturbation, $\delta \underline{z}$. Similarly, G maps the measurement noise vector, $\underline{\nu}$ to $\delta \underline{z}$. Appendix B contains the derivations for each of the sensor measurements and further discusses the validity of this linear approximation.

The H matrix is defined as follows

$$H\left(\hat{\underline{x}}\right) = \begin{bmatrix} H_{u,p} & H_c \end{bmatrix} \tag{5.5.2}$$

$$H_{u,p} = \begin{bmatrix} -D_p & 0_{2\times3} & D_{\theta_b^n} & 0_{2\times6} & D_{\theta_k^b} & 0_{2\times3} \\ 0_{2\times3} & 0_{2\times3} & -D_l & 0_{2\times6} & 0_{2\times3} & -D_l \\ I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times6} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{5.5.3}$$

$$H_c = \begin{bmatrix} D_p & 0_{2\times3} & D_{\theta_c^n} & 0_{2\times4} \\ 0_{2\times3} & 0_{2\times3} & D_l & 0_{2\times4} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times4} \end{bmatrix} \tag{5.5.4}$$

where the 3×3 block matrices D_p, $D_{\theta_b^n}$, $D_{\theta_k^b}$, $D_{\theta_c^n}$, and D_l are defined below. The $I_{3\times3}$ element denotes a 3×3 identity matrix.

$$H_{los} \triangleq \begin{bmatrix} \frac{1}{l_z^k} & 0 & -\frac{l_x^k}{\left(l_z^k\right)^2} \\ 0 & \frac{1}{l_z^k} & -\frac{l_y^k}{\left(l_z^k\right)^2} \end{bmatrix} \tag{5.5.5}$$

$$D_p \triangleq H_{los}\hat{R}_b^k\hat{R}_n^b$$

$$D_{\theta_b^n} \triangleq -D_p\left[\left(\hat{\underline{p}}_c^n\times\right) + \left(\hat{R}_c^n\underline{r}_f^c\times\right) - \left(\hat{\underline{p}}^n\times\right)\right]$$

$$D_{\theta_k^b} \triangleq -H_{los}\hat{R}_b^k\left[\left(\hat{R}_n^b\hat{\underline{p}}_c^n\times\right) + \left(\hat{R}_n^b\hat{R}_c^n\underline{r}_f^c\times\right) - \left(\hat{R}_n^b\hat{\underline{p}}^n\times\right)\right]$$

$$D_{\theta_c^n} \triangleq D_p\left(\hat{R}_c^n\underline{r}_f^c\times\right)$$

$$D_l \triangleq L_m\hat{R}_b^l\hat{R}_n^b\left(\hat{R}_c^n\underline{u}^c\times\right)$$

where $\underline{r}_f^c$ is the vector from the origin of the carrier frame to a fiducial of interest on the carrier.

The derivation of the above equations are shown in Appendix B.

5.6 Covariance Propagation

This section is dedicated to the propagation of the state covariances where equation 5.6.1 shows the governing differential equation

$$\dot{P} = F\left(\hat{\underline{x}}\right)P + PF^T\left(\hat{\underline{x}}\right) + BQB^T \tag{5.6.1}$$

Recall that F is the linearized system dynamics matrix, B is the process noise coupling matrix (equation 5.4.33), and Q is the noise Power Spectral Density (PSD), which is defined as a block diagonal matrix shown in equation 5.6.2.

$$Q = \begin{bmatrix} Q_\nu & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0 & 0 & 0 \\ 0_{3\times3} & Q_\omega & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0 & 0 & 0 \\ 0_{3\times3} & 0_{3\times3} & Q_a & 0_{3\times3} & 0_{3\times3} & 0 & 0 & 0 \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & Q_g & 0_{3\times3} & 0 & 0 & 0 \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & Q_{vc} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & Q_{\omega cx} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & Q_{\omega cy} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & Q_{\omega cz} \end{bmatrix} \tag{5.6.2}$$

with each of the diagonal terms defined as:

$$Q_\nu = VRW^2\,(I_{3\times3})$$

$$Q_\omega = ARW^2\,(I_{3\times3})$$

$$Q_a = \frac{2\sigma_{a,ss}^2}{\tau_a}\,(I_{3\times3})$$

$$Q_g = \frac{2\sigma_{g,ss}^2}{\tau_g}\,(I_{3\times3})$$

$$Q_{vc} = CVRW^2\,(I_{3\times3})$$

$$Q_{\omega cx} = 3.2e^{-6} \qquad \text{(see Section 6.3)}$$

$$Q_{\omega cy} = \frac{2\sigma_{cy,ss}^2}{\tau_{\omega cy}}$$

$$Q_{\omega cz} = \frac{2\sigma_{cz,ss}^2}{\tau_{\omega cz}}$$

The parameters used in the above equations were chosen to match typical values from sensor datasheets or calculated to fit the SCONE datasets. These values, as well as other parameters used in the verification of equation 5.6.1, are tabulated and explained in Section 6.3.

5.7 Kalman Update

The equations governing the Kalman update step are discussed in this section. The following equations are computed for each of the sensor measurements at every Kalman update.

The Kalman gain, K, is a weighting matrix that multiplies the measurement residuals to update the error state vector, $\delta\underline{\hat{x}}^{+}$. This relationship is shown in equation 5.7.1 where the Kalman gain is defined by equation 5.7.2.

$$\delta\underline{\hat{x}}^{+} = K\left[\underline{\tilde{z}} - \underline{h}\left(\underline{\hat{x}}^{-}\right)\right] \tag{5.7.1}$$

$$K = P^{-}H^{T}\left(\underline{\hat{x}}^{-}\right)\left[H\left(\underline{\hat{x}}^{-}\right)P^{-}H^{T}\left(\underline{\hat{x}}^{-}\right) + GRG^{T}\right]^{-1} \tag{5.7.2}$$

Recall that G is the matrix that maps the measurement noise, $\underline{\nu}$, to the residuals. The R matrix is a diagonal matrix of the measurement variance. The diagonal values for the R matrices are shown in Table 6.2.

The covariance is also updated using the Kalman gain as defined below in equation 5.7.3.

$$P^{+} = \left[I - KH\left(\underline{\hat{x}}^{-}\right)\right]P^{-}\left[I - KH\left(\underline{\hat{x}}^{-}\right)\right]^{T} + KGRG^{T}K^{T} \tag{5.7.3}$$

Using the equations discussed above, the state estimate vector, $\underline{\hat{x}}$, is then updated by correcting the errors. This is accomplished by using the updated error state vector and the mapping defined in 4.3.2 to create the a posteriori state estimate. At this point, the measurement residuals are updated and the a posteriori estimate becomes the a priori estimate for the next measurement.

5.8 Relative Covariance Transforms

Because this project focuses on a UAV/carrier landing scenario, the absolute position of either vehicle is not as important as the relative position from one to the other. It is thus necessary to compute the relative covariances for the position, velocity, and attitude between the two vehicles. This is accomplished with the following equations.

$$P_{rel,p} = A_p P A_p^T \tag{5.8.1}$$

$$P_{rel,v} = A_v P A_v^T \tag{5.8.2}$$

$$P_{rel,a} = A_a P A_a^T \tag{5.8.3}$$

where

$$A_p = \begin{bmatrix} I_{3\times3} & 0_{3\times18} & -I_{3\times3} & 0_{3\times10} \end{bmatrix} \tag{5.8.4}$$

$$A_v = \begin{bmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times18} & -I_{3\times3} & 0_{3\times7} \end{bmatrix} \tag{5.8.5}$$

$$A_a = \begin{bmatrix} 0_{3\times6} & I_{3\times3} & 0_{3\times18} & -I_{3\times3} & 0_{3\times4} \end{bmatrix} \tag{5.8.6}$$

These relative covariances will be further analyzed in Section 6.6.

CHAPTER 6

FILTER VALIDATION

The EKF is a sophisticated algorithm with many components. To produce valid and trustworthy results, it is necessary to confirm that the EKF behaves as expected. This is most conveniently accomplished by checking individual pieces of the EKF. This chapter presents a procedure for checking individual elements of the EKF implementation discussed in Chapter 5. Section 6.1 presents a method of checking the error state vector and the matrix mapping equations for consistency. The next section confirms the nonlinear state and measurement propagation. Section 6.4 validates the measurement residuals. The final two sections present the simulation parameters and Monte Carlo plots to validate the estimation capability.

6.1 Error State Vector Validation

The three matrix equations 4.3.2 - 4.3.4 provide a useful method for validating the state vectors in the EKF. Since these equations are used for validation, it is necessary to check for consistency between the equations themselves. This is accomplished by first manually adding errors to the estimated vector using equation 4.3.3. Then by applying equation 4.3.4, the estimated errors are calculated and verified that they are equal to the errors that were manually added in the previous step. The final check is to apply these errors in equation 4.3.2 to ensure that the output is equal to the initial truth state. The results of this validation step are recorded in Table 6.1

The $\delta\underline{x}$ column in Table 6.1 is the difference between the initial value of the state variable and the value after injecting and removing the errors via the mapping equations. The results of the table show that these equations are consistent and produce the expected output. The quaternions in the state vector exhibit a small difference rather than a perfect mapping because of the first-order approximation used in the perturbation equations.

Table 6.1: Mapping equation validation

Variable Name	$\delta\underline{x}$
$\underline{p}^n$	0.0
$\underline{v}^n$	0.0
q_b^n	1.110299228302966e-16
$\underline{b}_a$	0.0
$\underline{b}_g$	0.0
q_k^b	2.452840257200934e-18
q_l^b	1.936866724003709e-18
$\underline{p}_c^n$	0.0
$\underline{v}_c^n$	0.0
q_c^n	1.359739955510518e-16
$\omega_{cx/n}^c$	0.0
$\alpha_{cx/n}^c$	0.0
$\omega_{cy/n}^c$	0.0
$\omega_{cz/n}^c$	0.0

However, based on the magnitude of this error, the inconsistency is negligible.

6.2 Nonlinear State Propagation

This section discusses the results of the truth state propagation equations and validates that the measurement models previously described are consistent.

Using the simulation, a flight trajectory was generated that resembles a typical landing approach. The UAV begins at an altitude of 120 meters and for the first 30 seconds, performs an "S" maneuver to aid in the calibration of the EKF. For the next 30 seconds, the UAV follows a glide slope that leads the aircraft to a reference point on the ship deck. Figure 6.1 shows the flight trajectory used in the simulation.

In the absence of measurement noise, the residuals should be zero. Figures 6.2 - 6.3 show the measurement residuals for the camera and lidar. As shown in these plots, the

error has negligible magnitude indicating that the nonlinear state propagation equations are consistent and accurate. Figures 6.4 and 6.5 show perfect consistency for the GPS residuals and insignificant errors in the estimation of the UAV's position, velocity, and attitude states. The minimal errors in the residual plots indicate that the measurement models and estimates are consistent.

The plots for the UAV parameter states were left out of this section as they were identically zero for all time and provided no further insight. The plots have been included in Appendix C.1 for completeness.

At this stage, the carrier state estimates were replaced with the carrier truth data in order to get the measurement residual plots to behave as expected. Any misalignment in the carrier estimates would manifest themselves as a measurement error. To confirm that the measurements were working properly, the errors in the carrier estimate were zeroed out for this validation step and thus their corresponding plots were not included.

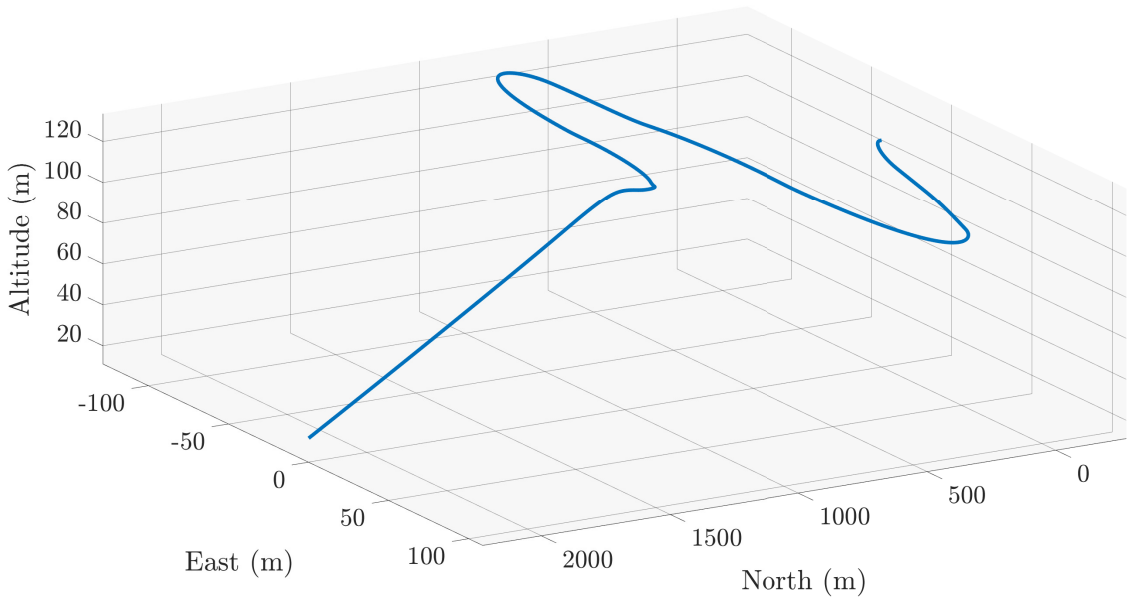

Fig. 6.1: UAV flight trajectory

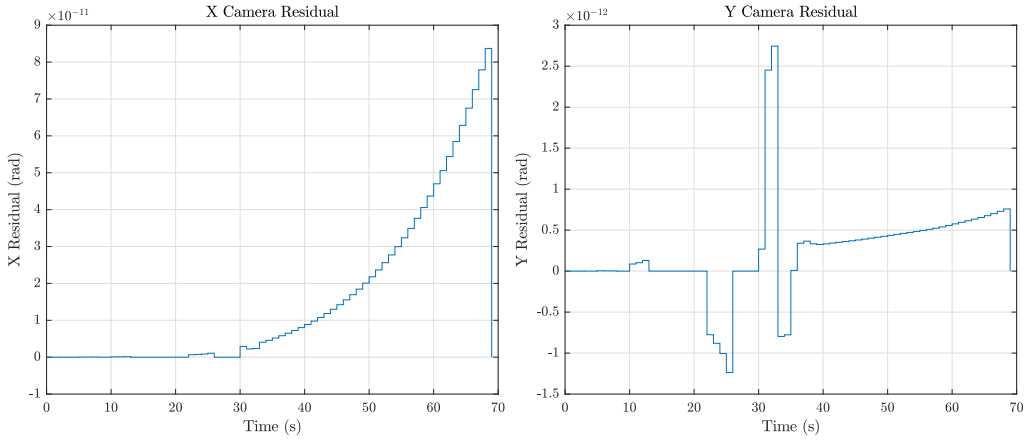

Fig. 6.2: Camera residual propagation

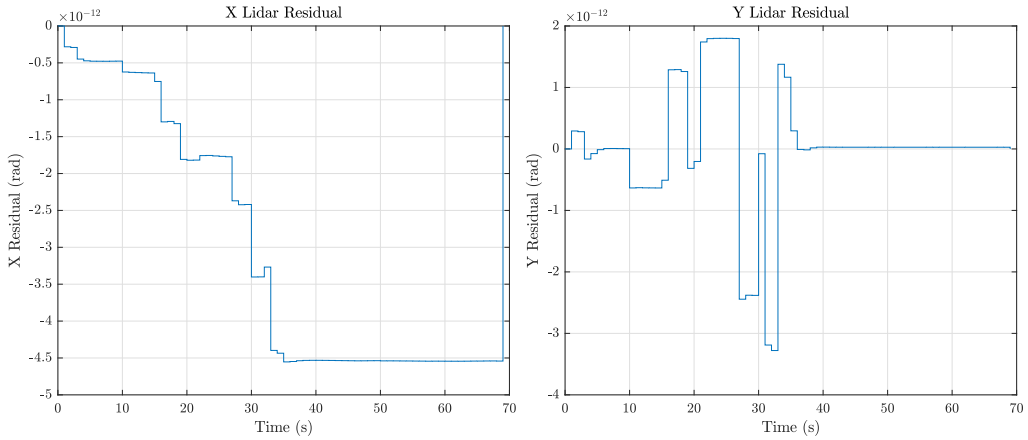

Fig. 6.3: Lidar residual propagation

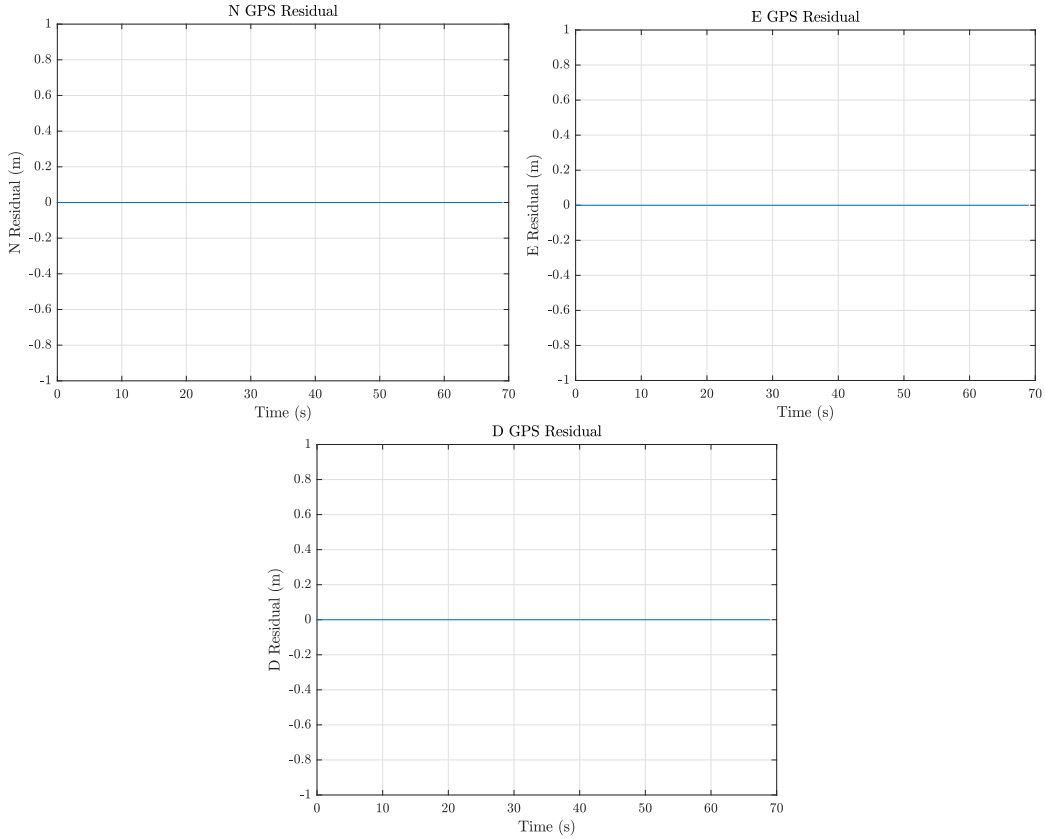

Fig. 6.4: GPS residual propagation

6.3 Simulation Parameters

To verify that the error state covariance propagation is correct, a Monte Carlo simulation was implemented. After running 125 cycles of the simulation, the propagation of the error covariance was compared with the ensemble statistics of the error propagated using the nonlinear equations presented in Section 5.1. This simulation requires many tuning parameters with the values of these parameters displayed in the following tables.

Table 6.3 presents the UAV's accelerometer parameters while Table 6.4 shows the corresponding parameters for the gyroscope. Both sensors were modeled as a first order Markov process and therefore require a time constant for the sensor bias, a standard deviation of the steady-state value, and a random-walk noise strength. Table 6.5 contains the carrier's parameters as recorded in SCONE datasets. These values were determined by plotting the

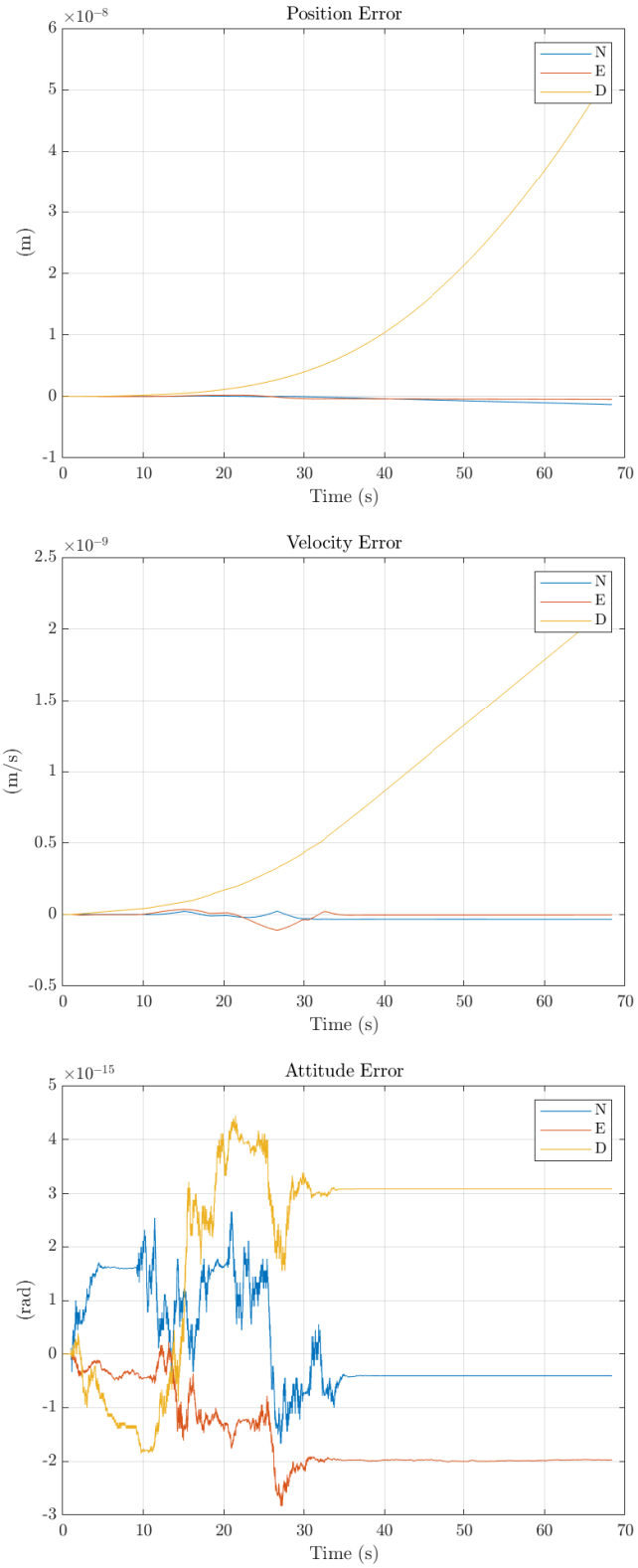

Fig. 6.5: UAV state residual propagation

Table 6.2: Measurement parameters

Parameter	X	Y	Z	Units	Description
σ_{cam}	0.007	0.007	-	rad	3σ camera uncertainty
σ_{lidar}	0.005	0.005	-	rad	3σ lidar uncertainty
σ_{gps}	1.0	1.0	3.0	m	3σ GPS uncertainty

Table 6.3: Accelerometer parameters

Parameter	Value	Units	Description
τ_a	60.0	sec	Accelerometer bias time constant
$\sigma_{a,ss}$	0.001	g	3σ steady-state accelerometer bias
VRW	0.06	$m/s/\sqrt{hr}$	3σ velocity random walk

PSD of the carrier's attitude and fitting a curve to the data. In the case of the carrier's x-axis, a second order equation was required to accurately model the behavior. The magnitude of the PSD for the y and z axes was significantly smaller than the x-axis and thus a first order Markov was chosen as an approximation for these two axes. This PSD analysis can be seen in Figure 6.6.

The final Table, 6.6, displays the Kalman filter's initial uncertainty values for the given states. These values were used to initialize the covariance propagation equation discussed in Section 5.6. Most of these values were chosen based off of similar values found in [24] or typical tactical grade sensors such as [25]. The camera and lidar are assumed to be mechanically aligned and therefore have very small misalignment errors. The carrier values were chosen to be conservative to observe the covariances clamping down as a result of the EKF update.

Table 6.4: Gyroscope parameters

Parameter	Value	Units	Description
τ_g	60.0	sec	Gyroscope bias time constant
$\sigma_{g,ss}$	1.0	deg/hr	3σ steady-state gyroscope bias
ARW	0.07	$deg/\sqrt{hr}$	3σ angle random walk

Table 6.5: Carrier parameters

Parameter	Value	Units	Description
CVRM	0.045	$m/s/\sqrt{hr}$	1σ carrier velocity random walk
Q_{cx}	$3.2e^{-6}$	rad^2/s^3	Carrier roll PSD strength
ζ_{cx}	0.0200	unitless	Carrier roll damping ratio
ω_{ncx}	0.0503	Hz	Carrier roll cut-off frequency
τ_{wcy}	1.000	sec	Carrier attitude rate time constant about y-axis
$\sigma_{cy,ss}$	$3.7e^{-4}$	rad/s	1σ steady-state carrier attitude rate about y-axis
τ_{wcz}	1.000	sec	Carrier attitude rate time constant about z-axis
$\sigma_{cz,ss}$	$7.1e^{-4}$	rad/s	1σ steady-state carrier attitude rate about z-axis

Table 6.6: State initial uncertainty values

Parameter	X	Y	Z	Units
UAV position	1.00	1.00	3.00	m
UAV velocity	0.10	0.10	0.10	m/sec
UAV attitude	0.10	0.10	0.10	rad
Accelerometer bias	0.001	0.001	0.001	g
Gyroscope bias	1.00	1.00	1.00	deg/hr
Camera attitude	0.001	0.001	0.001	rad
Lidar attitude	0.001	0.001	0.001	rad
Carrier position	30.0	10.0	10.0	m
Carrier velocity	6.00	1.00	1.00	m/sec
Carrier attitude	0.20	0.20	0.20	rad
Carrier angular velocity	0.20	0.20	0.10	rad/s
Carrier angular acceleration	0.0001	-	-	rad/s^2

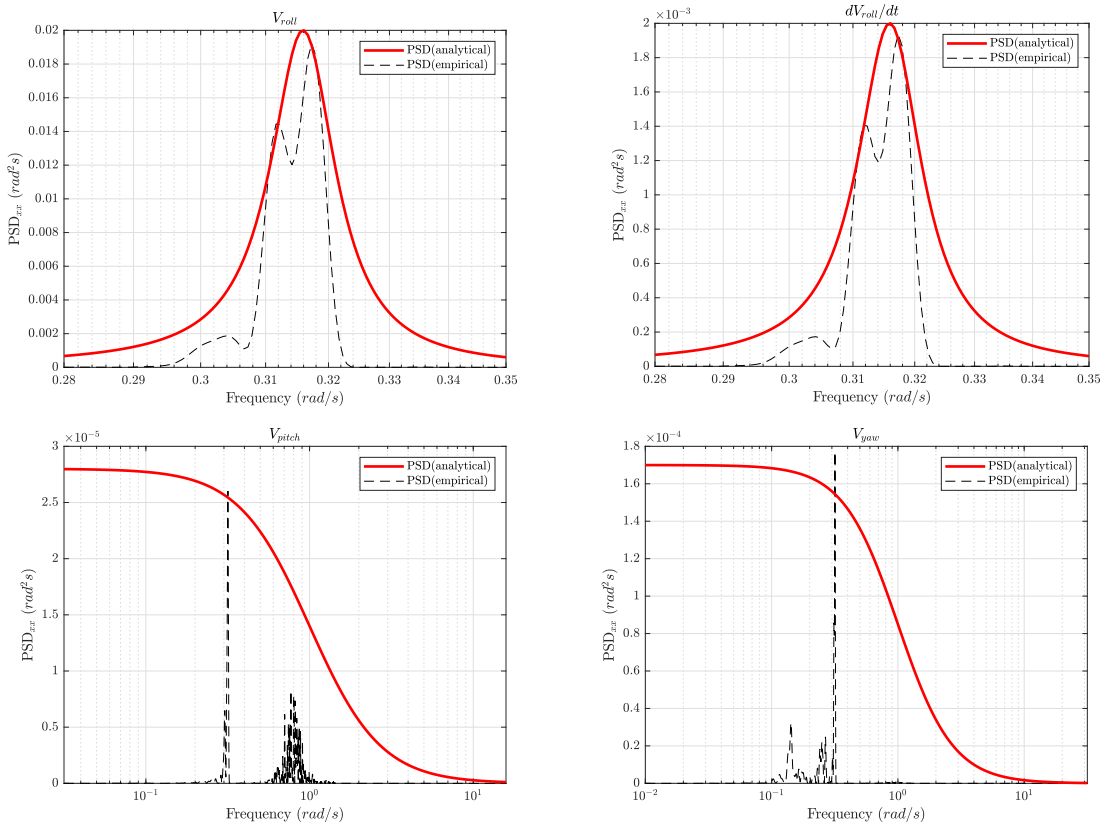

Fig. 6.6: Carrier attitude PSD analysis

6.4 Residual Verification

At this point, it is necessary to validate that the residuals behave as expected. Each of the measurement residuals should be zero-mean, white noise, and have a covariance consistent with the residual covariance matrix. To verify these characteristics are met, a single Monte Carlo simulation was ran with the parameters listed in Tables 6.3 - 6.6.

Figures 6.7 - 6.9 show the measurement residuals after a single Monte Carlo run. The camera residuals were only processed when the fiducials on the ship deck were within the field of view of the onboard camera. At the beginning of the simulation when the UAV is performing the "S" maneuver, there are certain times when the camera is unable to see the fiducials. Thus Figure 6.7 shows no residual at these times.

The figures show that the residuals have all the expected characteristics and are therefore valid to be used in the Kalman update step.

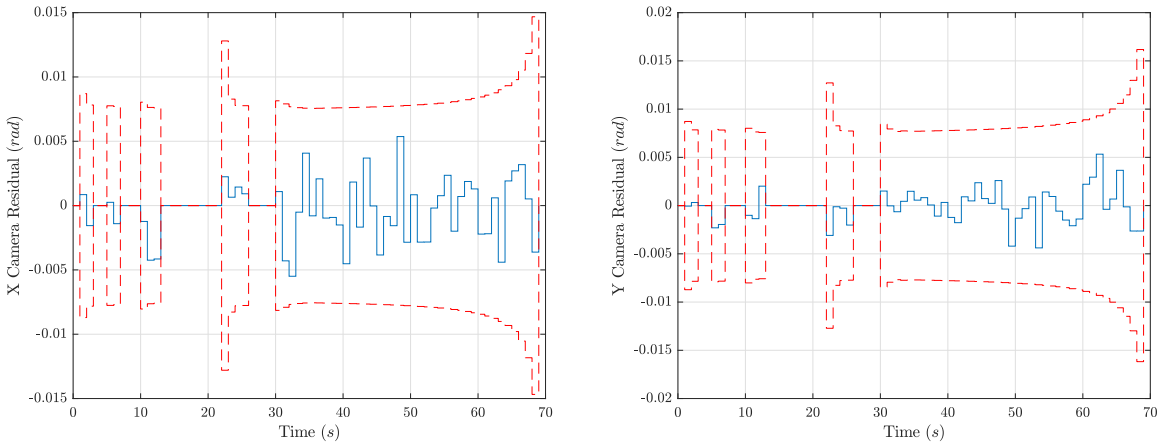

Fig. 6.7: Camera residuals

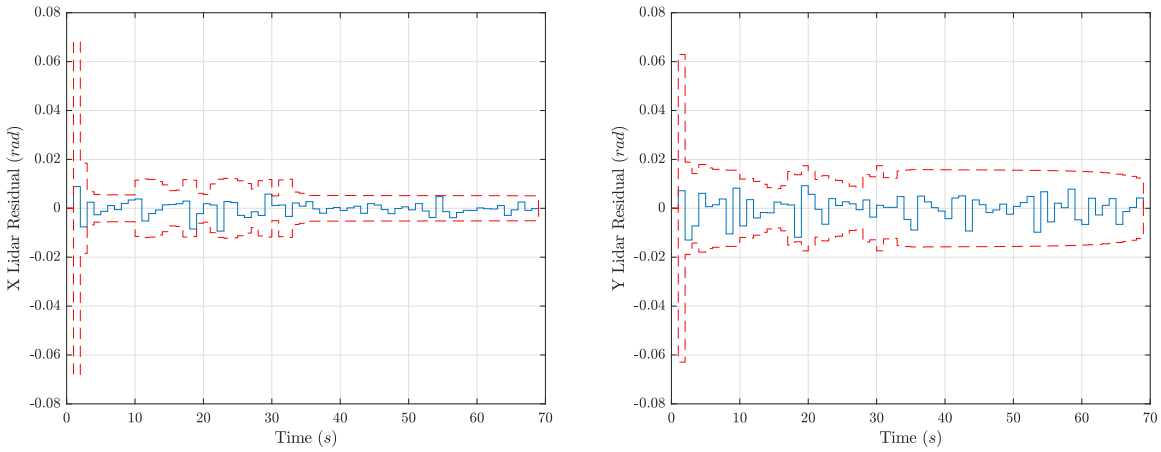

Fig. 6.8: Lidar residuals

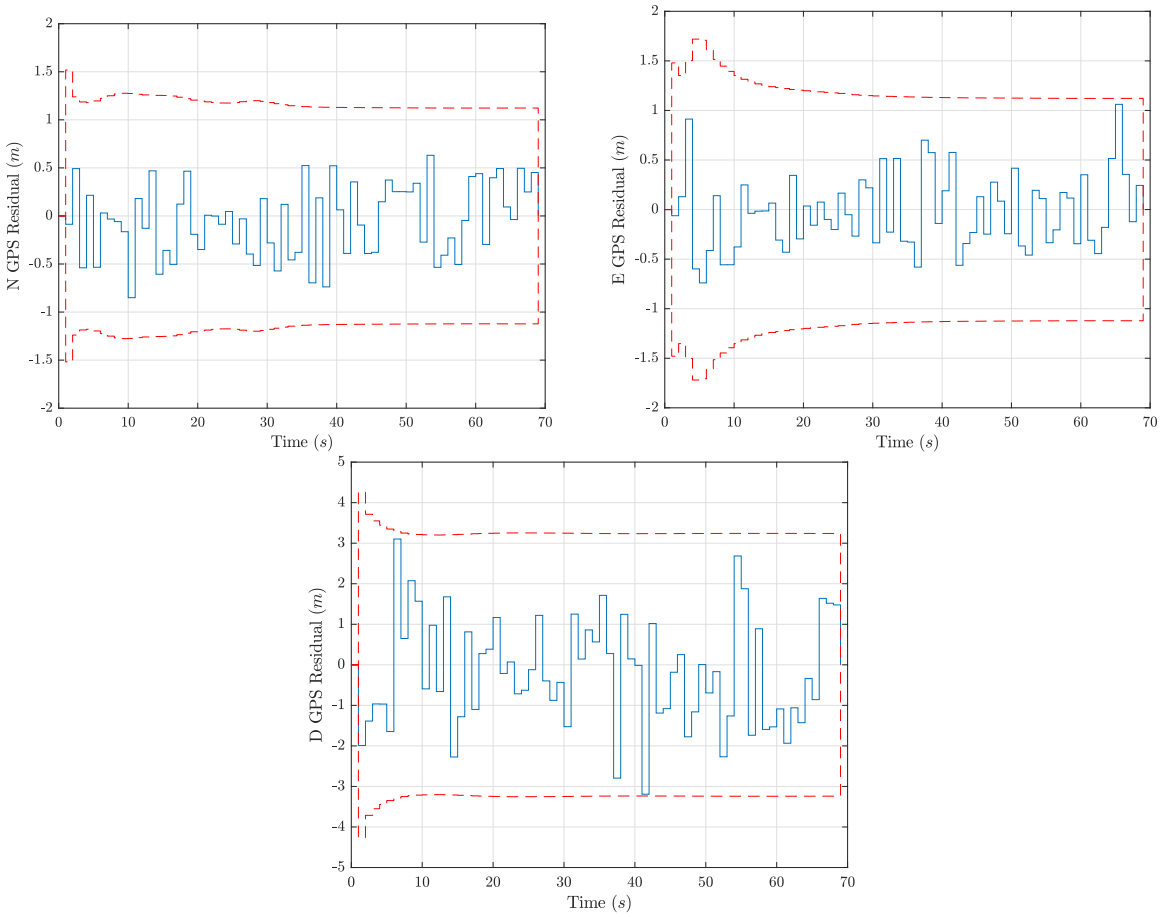

Fig. 6.9: GPS residuals

6.5 Covariance Propagation

For small errors, the propagation of the error covariance using equation 5.6.1 should match the ensemble statistics of errors propagated through the nonlinear equations. Figures 6.10 - 6.12, show "hairline" plots for the UAV and carrier position, velocity, and attitude states. The red dotted-lines are the $\pm 3\sigma$ of the estimation errors, and each gray "hair" is a single Monte Carlo run. These plots are used to verify that the ensemble statics are consistent with the propagated covariance and are zero-mean.

To create variability in the carrier's states, each Monte Carlo run started at a random index in the SCONE dataset. The values at this random index were then subtracted off from the carrier's states to ensure that at the end of the simulation the UAV would land

on the carrier regardless of the starting index in the SCONE data.

Each of the UAV plots produced expected results with the large percentage of hairs staying within the 3σ bounds. The down-velocity subfigure does show some inconsistency as the mean of the hairs is slightly biased. This bias can also be seen in the down-position subfigure since the position is directly related to the UAV's velocity.

In general, the carrier state plots are more conservative than the UAV state plots. This can been seen where the errors are slightly smaller than the 3σ bounds. The inconsistencies in the carrier states are likely due to the approximation of the chosen equations to model the carrier's behavior as recorded in the SCONE data files. However, since the covariance values are conservative, these approximations do not present any problems once the Kalman update is active. This can be seen in the Section 6.6.

Since the carrier's angular acceleration about the x-axis is a second order system, it has a natural frequency which is apparent in the oscillating behavior found in the carrier's North attitude subfigure in Figure 6.12. This behavior propagates upward, influencing the carrier's velocity and attitude plots which are consistent with the expected behavior for this axis. This angular acceleration plot as well as the remaining state plots are included in Appendix C.2.

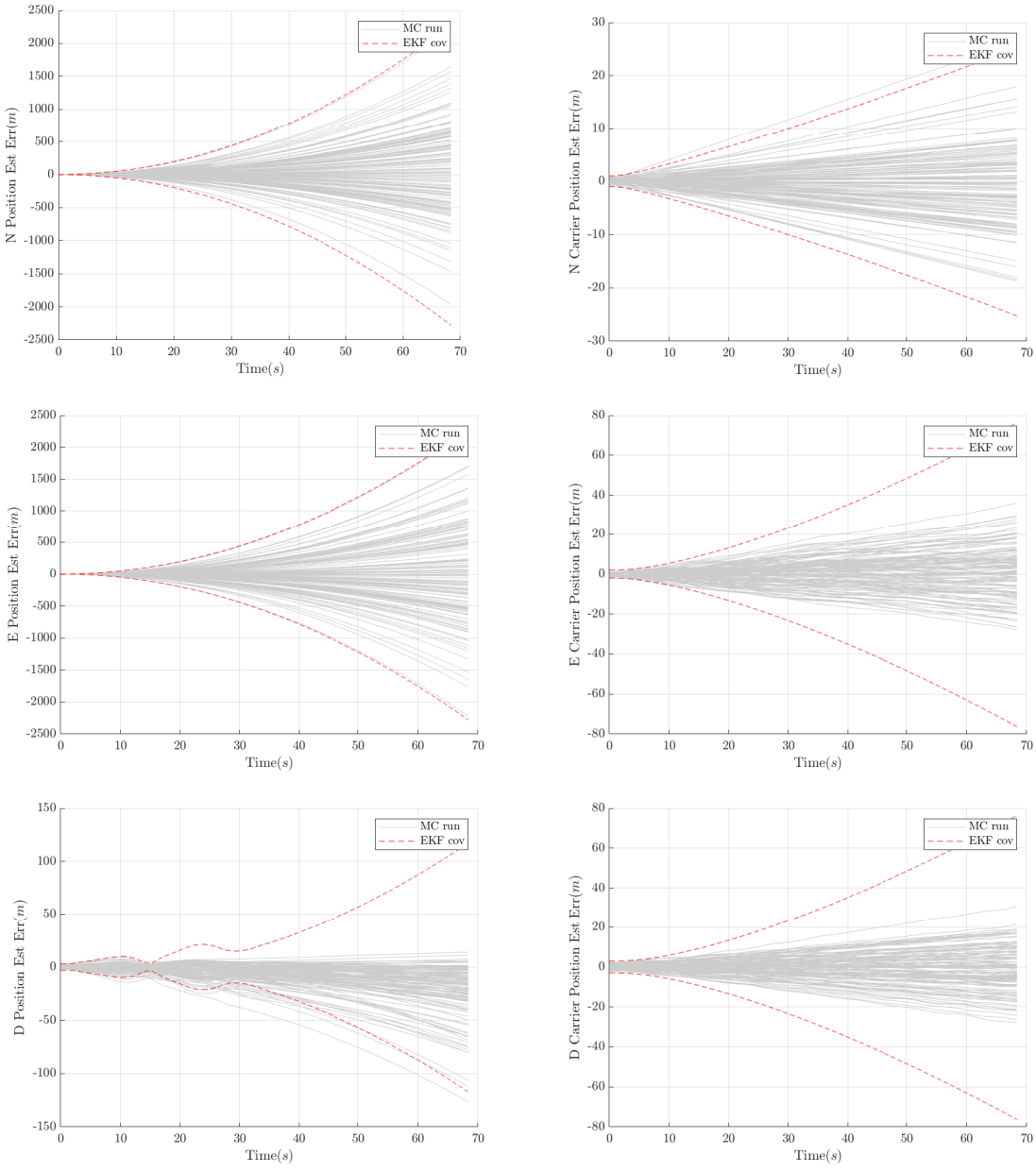

Fig. 6.10: UAV and carrier positions

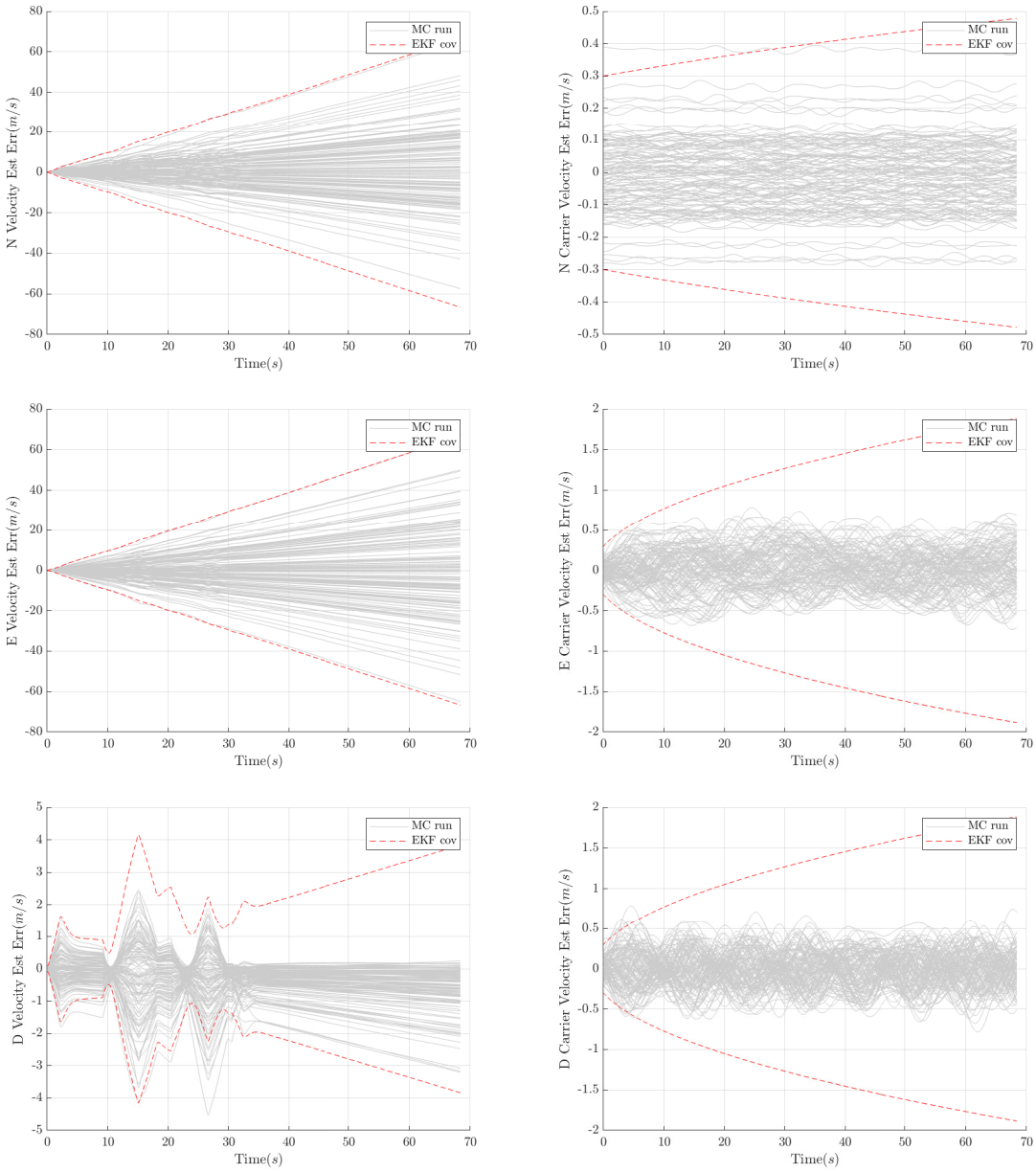

Fig. 6.11: UAV and carrier velocities

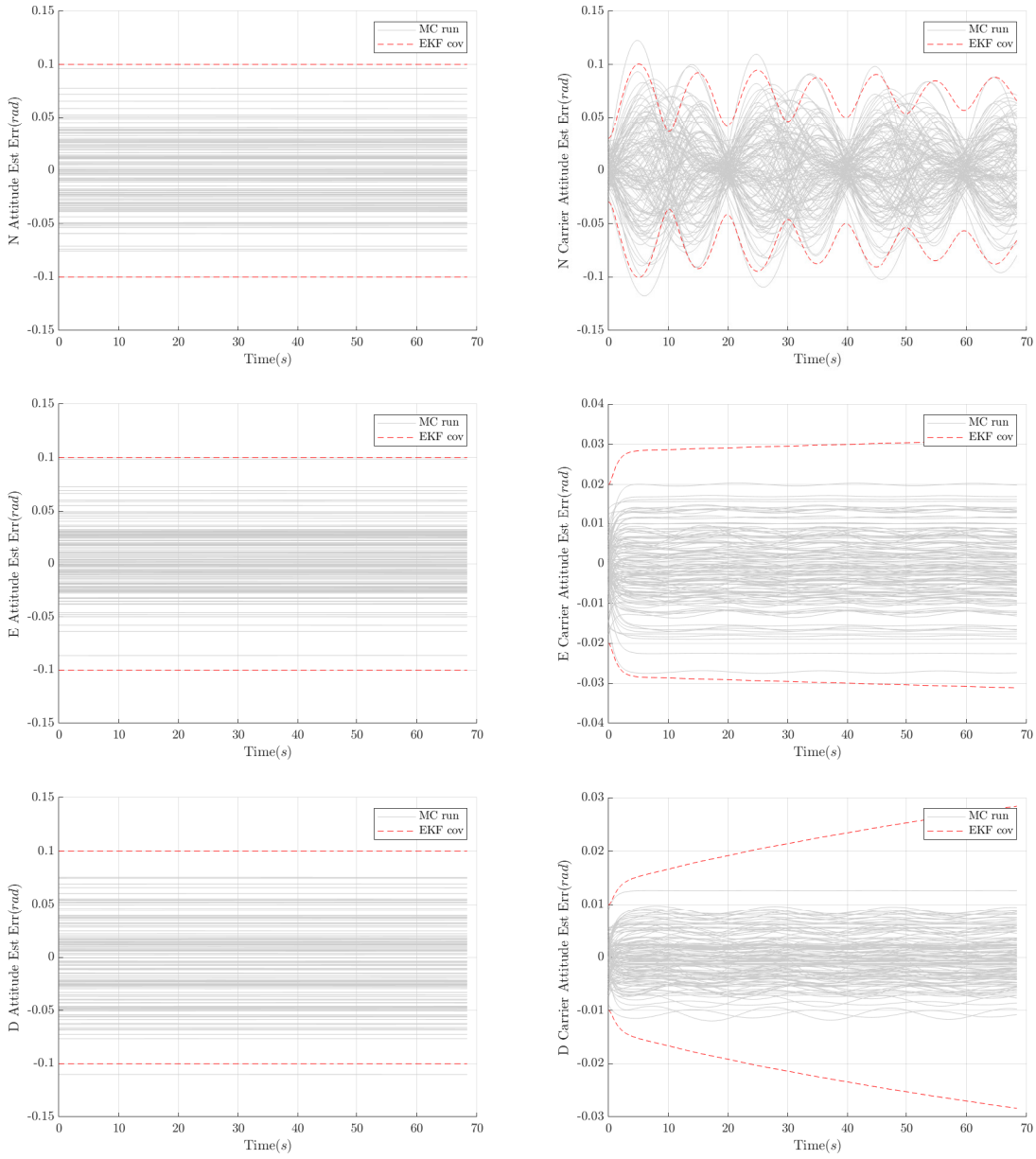

Fig. 6.12: UAV and carrier attitudes

6.6 State Estimation

This section shows the hairline plots for each of the states with the Kalman filter processing sensor measurements. The Monte Carlo simulation was set to run 125 cycles with each hairline representing the estimation error for the corresponding state. This section focuses on the relative position, velocity, and attitude states from the UAV to the carrier. This is because the position of the carrier relative to the oncoming UAV carries more importance in a landing scenario than the absolute position of either vehicle. Appendix C.3 contains the remaining plots for completeness.

The majority of hairlines stay within the 3σ bounds for all of the relative states. There are a few hairs that stray out of bounds in the position and velocity plots, but with the introduction of the lidar, the relative attitude plots tighten down extremely well.

Since the velocity directly influences the carrier's position, the inconsistency in these plots is likely caused by a common inadequacy in the way the carrier's velocity was modeled. This is potentially due to an over simplification that was made when modeling the carrier's velocity contained in the dataset. However, even with the inconsistency, the filter covariance adequately represents the ensemble statistics of the estimation errors for each relative state.

Table 6.7 shows two carrier parameters that were adjusted once the Kalman update step was active. These values were tuned such that the 3σ bounds accurately represented the covariances. Previously, $Q_{cx,truth}$ and $Q_{cx,nav}$ were set to the same value which can be seen in Table 6.5.

Table 6.7: State estimation parameters

Parameter	Value	Units	Description
$Q_{cx,truth}$	$3.2e^{-5}$	rad^2/s^3	Carrier roll PSD strength
$Q_{cx,nav}$	$8.0e^{-4}$	rad^2/s^3	Carrier roll PSD strength

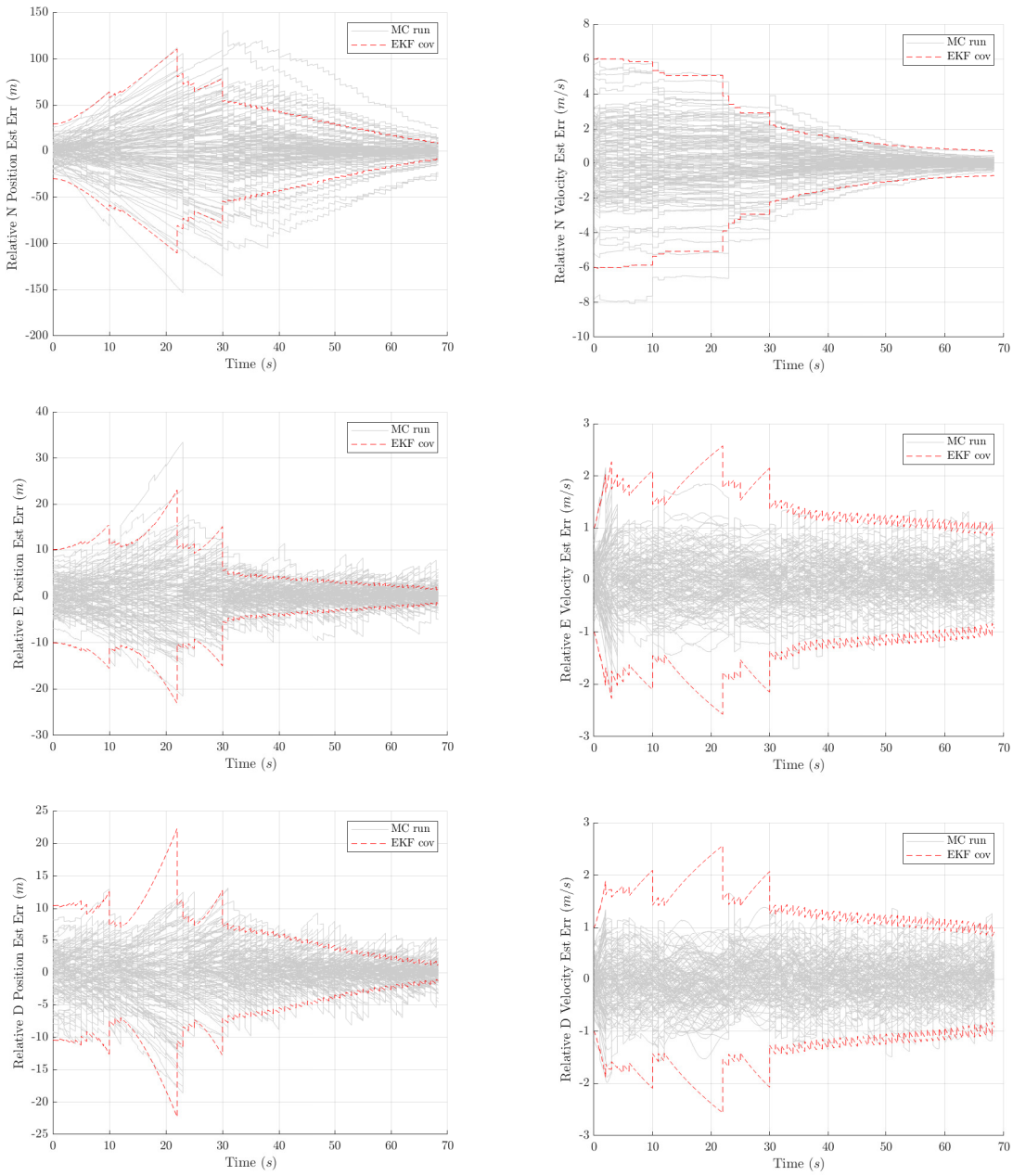

Fig. 6.13: Relative position and velocity estimate

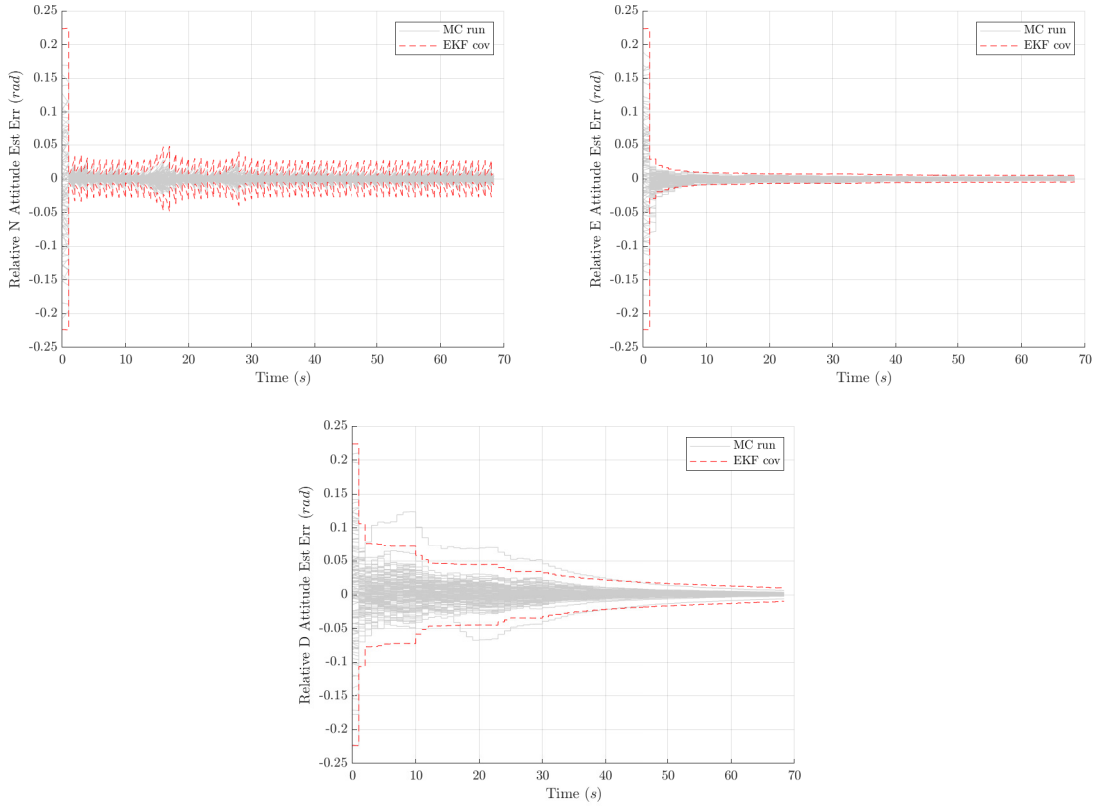

Fig. 6.14: Relative attitude estimate

CHAPTER 7

RESULTS

This chapter is dedicated to the results pertaining to the design objectives presented in the introduction. With the first objective complete and verified as discussed in Chapters 5 and 6, the primary focus in this chapter is to present the results of the navigation error analysis. This chapter includes plots for the Root Sum Square (RSS) of the three components in each relative state. The RSS covariance is defined as:

$$\sigma_{RSS} \triangleq \sqrt{(\sigma_x)^2 + (\sigma_y)^2 + (\sigma_z)^2} \tag{7.0.1}$$

For completeness, Appendix D includes the plots for each component in every relative state whereas this chapter focuses primarily on the RSS of each relative state.

This chapter is split into two sections. The first will focus on the navigation sensitivity to onboard sensor grades. The second section will focus on the sensitivity to environmental factors. This includes the amount of time without a GPS lock and the number of fiducials on the ship deck.

7.1 Sensor Grade Sensitivity

This section presents the results of the navigation sensitivity to onboard sensor grades. The same general procedure was followed for each study. Three separate instances of the simulation were ran with a differing magnitude of errors in the sensor. The error magnitude is representative of the sensor's quality and measurement accuracy and ranges from low to high grade. In regard to the IMU, the grades low, medium, and high correspond to consumer, tactical, and navigation grades respectively. During a specific sensor study, the measurements from the remaining sensors were active and processed as normal with the medium grade sensor values. For the sensor grade sensitivity studies, the GPS was disabled after 30 seconds to simulate the potential loss of position measurements prior to

Table 7.1: Sensor grade specifications

	Parameter	Low Grade	Medium Grade	High Grade	Units
IMU	$\sigma_{a,ss}$	0.01	0.001	0.0001	g
	VRW	0.6	0.06	0.006	$m/s/\sqrt{hr}$
	$\sigma_{g,ss}$	10.0	1.0	0.1	deg/hr
	ARW	0.7	0.07	0.007	$deg/\sqrt{hr}$
Camera	$\sigma_{cam,x}$	0.07	0.007	0.0007	rad
	$\sigma_{cam,y}$	0.07	0.007	0.0007	rad
Lidar	$\sigma_{lidar,x}$	0.05	0.005	0.0005	rad
	$\sigma_{lidar,y}$	0.05	0.005	0.0005	rad

the UAV landing. Table 7.1 contains the parameters for each onboard sensor and grade where the medium grade column contains the values used for filter validation in Chapter 6. The medium grade IMU values were based off of a typical tactical grade device, [25]. The medium grade camera values were calculated based on the number of pixels in an HD resolution image. The lidar values are highly dependent upon the plane fitting algorithm used in the lidar measurement. Since the implementation of this algorithm is not in the scope of this paper, these values were chosen similar to the values calculated for the camera. In every case, the low grade sensor values are a magnitude higher than the medium grade while the high grade values are a magnitude lower to represent a broad range of device fidelity.

With the EKF validated, the analysis parameters were varied to assess their effect on the state covariances. In this work, the state covariances 3 seconds prior to touchdown were selected as a reference time which is denoted by the vertical line at approximately 65 seconds in the following plots. The covariances at this reference time in the simulation will be the primary comparison value.

7.1.1 IMU Grade

This subsection analyzes the effect of IMU grade on the estimation covariances. Figures 7.1 and 7.2 show the RSS relative position, velocity, and attitude of the UAV and carrier. In all three plots, the difference between a tactical and a navigation grade IMU is negligible. The consumer grade IMU performs slightly worse than the others when estimating the relative position and velocity once the GPS signal is lost. As the UAV continues to approach the carrier, the covariances start to converge. At 3 seconds prior to landing, the position covariances are similar for all three grades. However, the velocity covariances at the reference time are improved slightly when using a navigation grade IMU over a consumer grade.

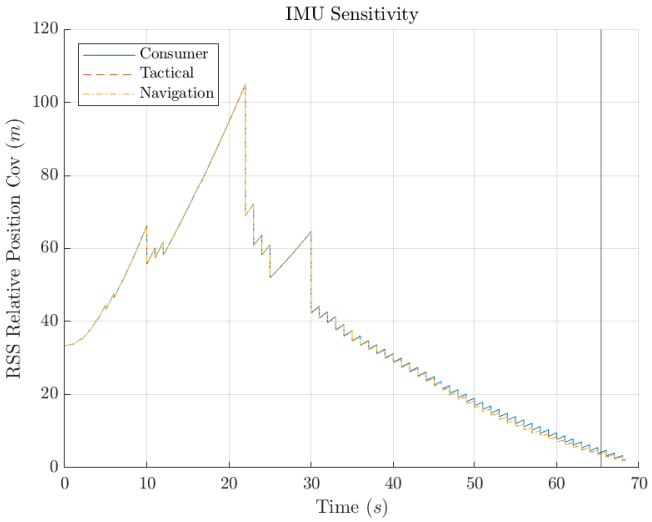

Fig. 7.1: IMU study - RSS relative position

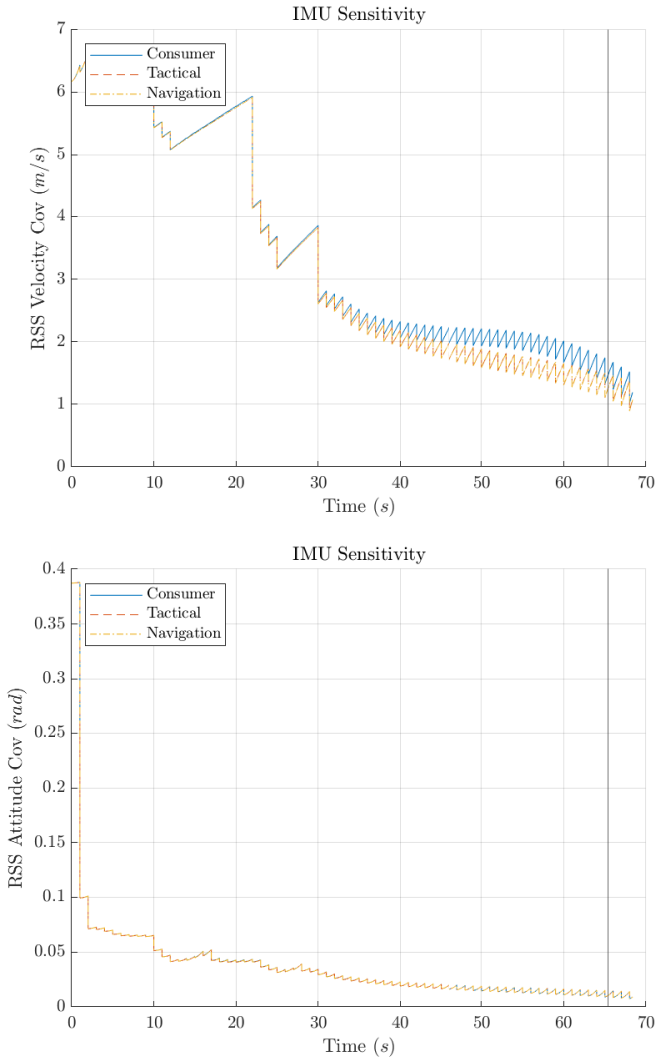

Fig. 7.2: IMU study - RSS relative velocity and attitude

7.1.2 Camera Errors

This subsection analyzes the effect of camera grade on the estimation covariances. Figure 7.3 shows the RSS relative position, velocity, and attitude states. As expected, having a higher fidelity camera produces smaller relative covariances. In contrast to the IMU study, the high grade has a significant improvement over both the medium and low grade cameras. In the case of the relative position, the covariances improve by roughly a factor of 60 at the reference time from the low grade to the high grade camera. This is a large improvement and is highly dependent upon the magnitude of errors in the camera. It can also be seen in the relative attitude plot that the camera grade has a significant affect on the covariances. A similar result can be seen by the lidar measurement which is primarily used for attitude determination. This effect will be further analyzed in Section 7.2.2.

7.1.3 Lidar Errors

Since the lidar measurement is used solely for the carrier attitude determination, the RSS relative position and velocity covariances are practically identical between the different lidar grades. The difference between a medium and high grade lidar is rather insignificant. The difference between high and low is a bit more dramatic with an improvement of a factor of 3. Figure 7.4 shows the results of this study.

Even with an improvement in covariances from low to high grade, it can be seen that this particular system architecture is largely insensitive to lidar errors. However, it is shown in Section 7.2.2 that having a lidar onboard significantly reduces the attitude covariances and reduces the number of required fiducials placed on the ship deck.

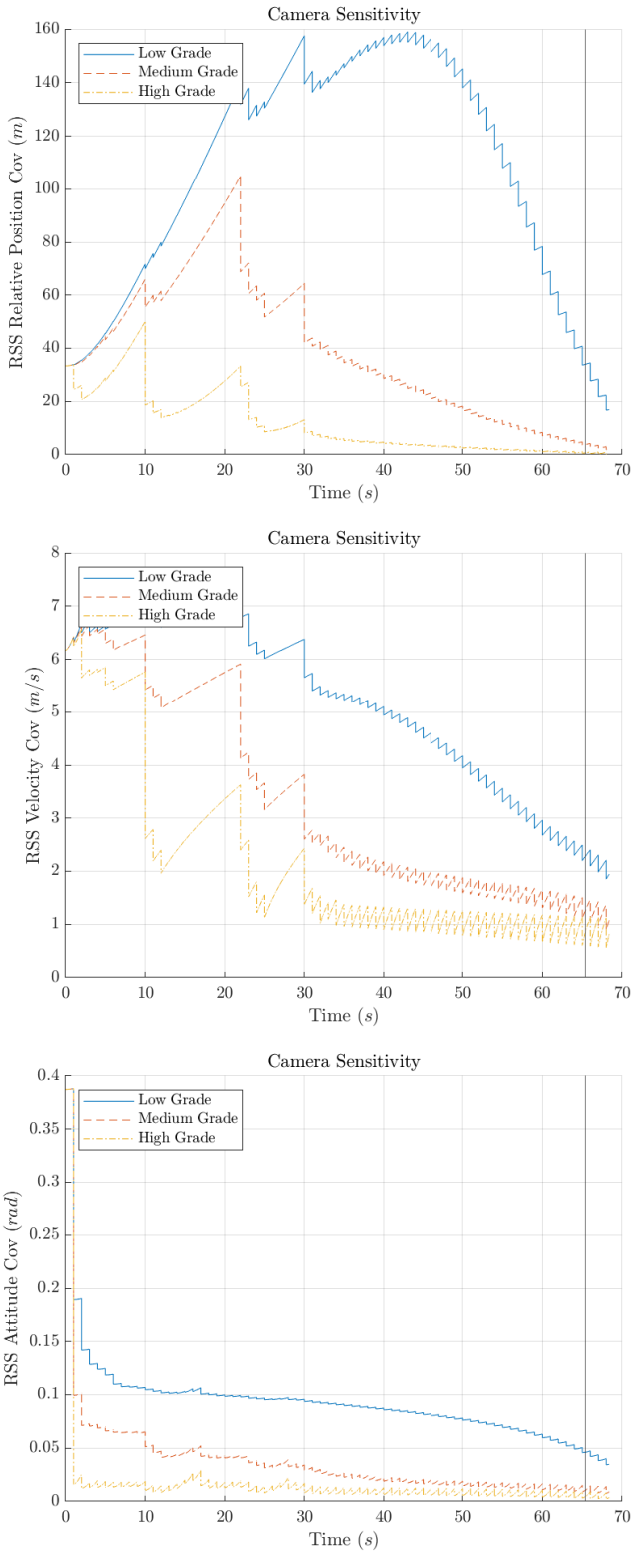

Fig. 7.3: Camera study - RSS relative position, velocity, and attitude

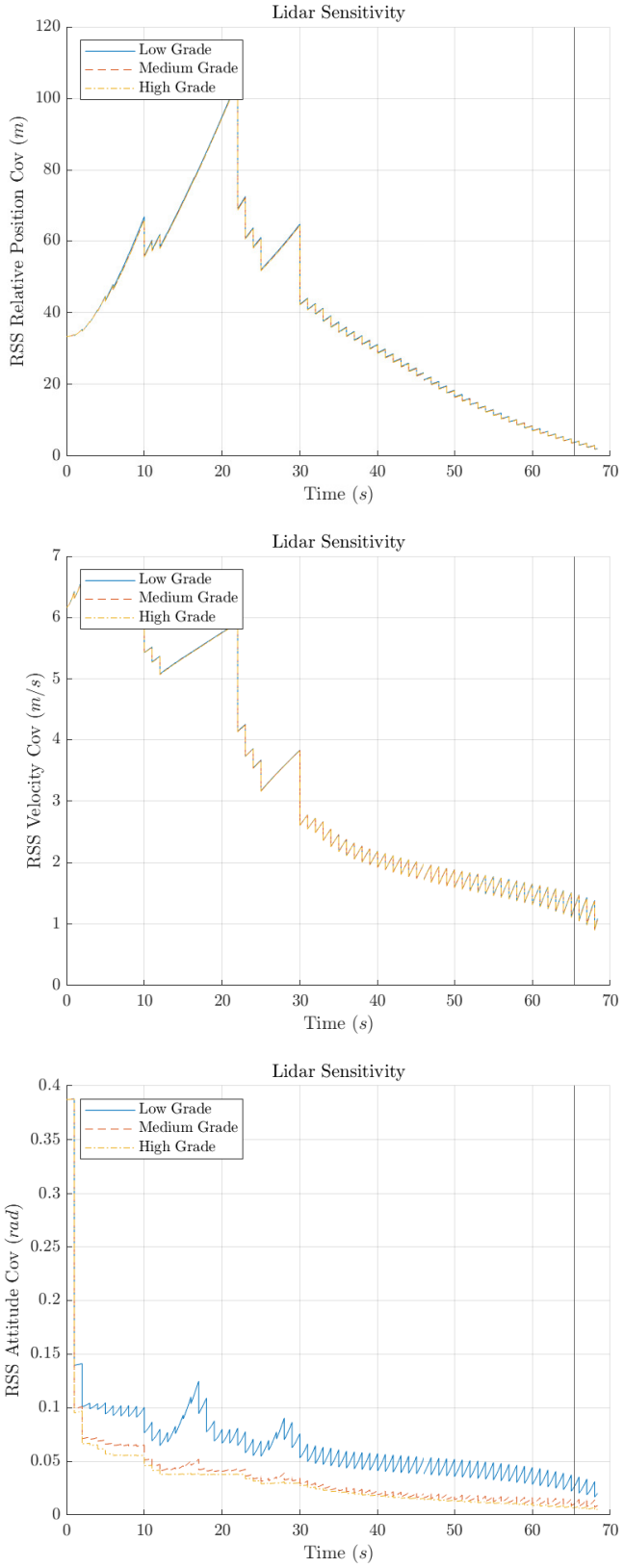

Fig. 7.4: Lidar study - RSS relative position, velocity, and attitude

7.2 Environmental Sensitivity

This section analyzes the navigation sensitivity to GPS-denied environments. The first study addresses the sensitivity of the system to the amount of time without a GPS signal. This shows how dependent the system is on position measurements. The second study covers the system's navigation errors while varying the number of fiducials placed on the ship deck.

7.2.1 GPS-denied

This section processes all onboard sensor measurements, but varies when the UAV loses GPS signal. Figure 7.5 shows the results of this study. As expected, the longer the UAV goes without position measurements, the larger the absolute position covariances become for both the UAV and the carrier. A similar trend is also seen in the relative velocities. The attitude plot shows the same behavior, but the magnitude on the covariances is small enough that the effect is arguably negligible. It should be noted that in all three cases, even though the absolute covariances increase as the amount of time without GPS increases, the relative covariances stay constant. In spite of being highly sensitivity to GPS errors for absolute position, the relative position between the UAV and carrier is largely unaffected. This is an important characteristic for a reliable and robust landing system.

7.2.2 Fiducial Number

This study focuses on determining the sensitivity to the number of fiducials on the ship deck. It can be seen in Figure 7.6, that the more fiducials placed on the deck, the smaller the relative position, velocity, and attitude covariances. This study was performed first with the camera only, and a second time with the lidar enabled. In the case of the relative position, the system requires three or more fiducials to achieve the same covariances as using the lidar. The relative velocity covariances become equal when at least two fiducials are used. The lidar has a much more dramatic improvement for the relative attitude. In fact even when using six fiducials on deck, the camera-only covariances never reach the same value as the system with the lidar.

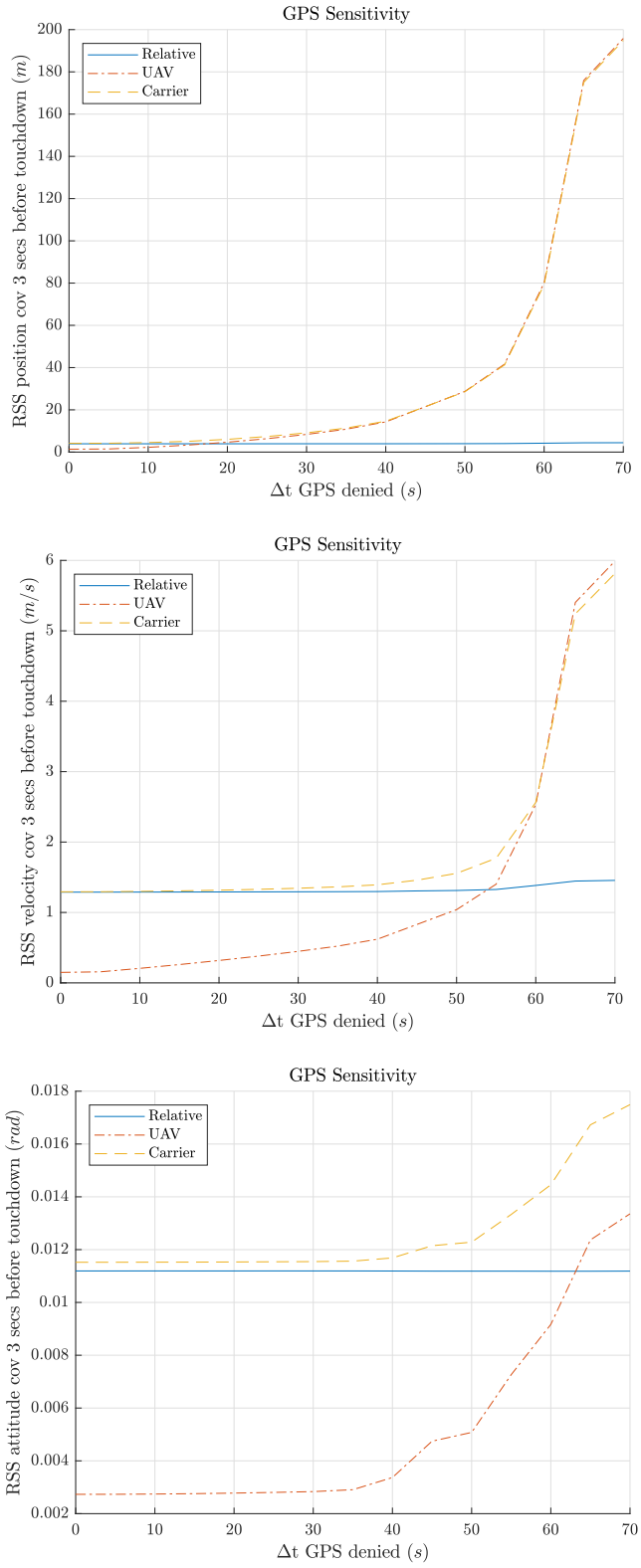

Fig. 7.5: GPS study - RSS relative position, velocity, and attitude

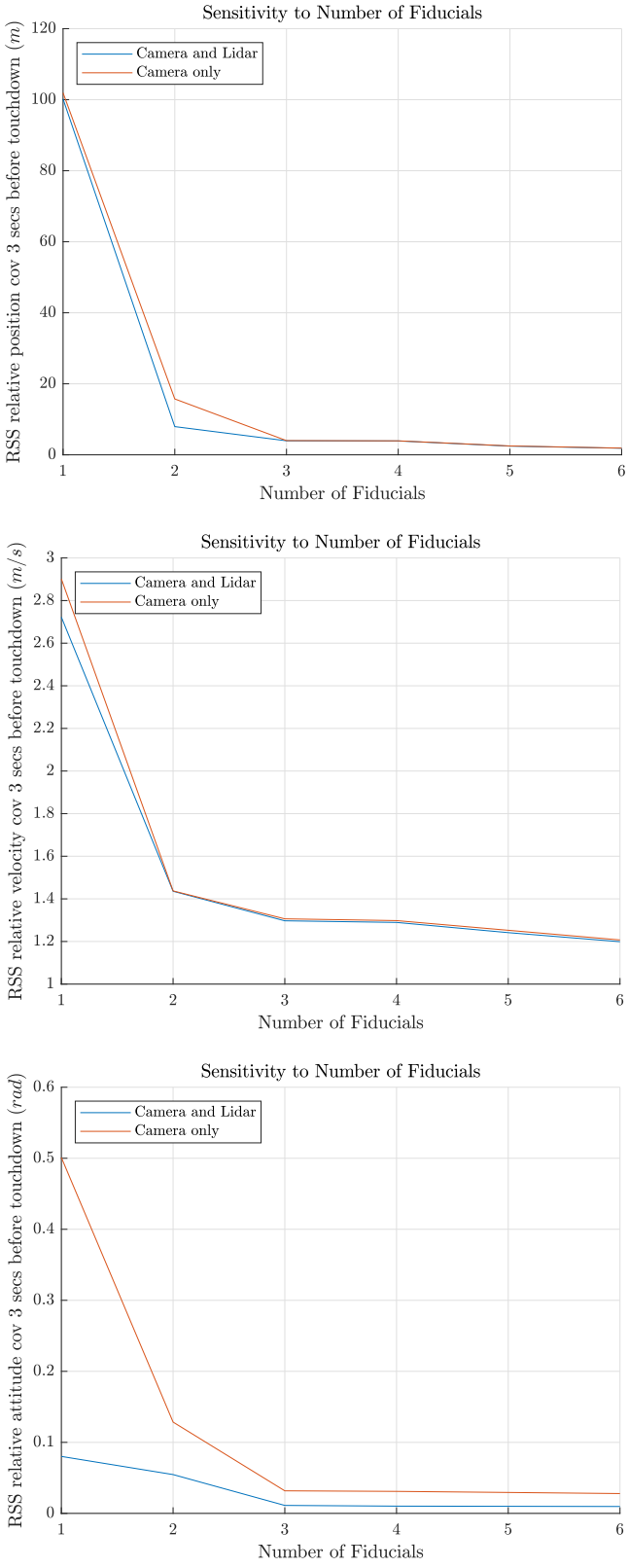

Fig. 7.6: Fiducial study - RSS relative position, velocity, and attitude

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Conclusion

In this paper a system-level analysis of UAV carrier landing sensitivities is discussed. Background information and the current state of the art in autonomous landing for UAVs is briefly reviewed. A common sensor configuration is chosen to represent UAVs in a carrier landing scenario with passive fiducials placed on the carrier deck. The sensitivity of this system to sensor errors and environmental factors is studied to determine tolerable errors in navigation.

In particular, the sensitivity of the relative UAV/carrier state covariances are analyzed as a function of IMU grade, camera errors, and lidar errors. In the scope of this work, environmental factors includes two cases. The first is the sensitivity of the system to GPS-denied environments where the amount of time without a GPS signal is varied to determine the system's sensitivity and dependence on position measurements. The second case focuses on the system's navigation errors while varying the number of fiducials on the ship deck.

From the simulation results, it is shown that the relative position and velocity of the UAV and carrier are more sensitive to errors in the camera than the onboard IMU. The system is largely insensitive to lidar errors, however having a lidar onboard significantly decreases attitude covariances and reduces the number of required fiducials placed on the ship deck. The study also shows that even in GPS-denied environments, the system is able to keep the relative covariances low which is an important characteristic for a reliable and robust landing system.

With the results from the analysis and application specific requirements, designers can determine tolerable errors and choose appropriate sensors to create systems at a lower cost without compromising landing accuracy.

8.2 Future Work

In most applications involving a lidar, the lidar is used to produce range measurements. In this work it was used solely for attitude determination. A possible extension to this work could extract more information from the lidar including range measurements to the ship deck as well as aid in determining the UAV's altitude along the landing approach. This would lead to a more accurate estimate of the relative position for precise landings.

Further research could also be conducted in determining the optimal placement of fiducials on the carrier. In this work, an attempt was made to strategically place the fiducials on the deck. Throughout the work it became apparent that not all fiducials configurations resulted in the same spread of covariances. Further research in this area would provide more insight for fiducial placement and number of required fiducials to keep relative covariances low for a safe and accurate landing.

Another potential study could include a control law to guide the aircraft to the desired location on the carrier. This would further validate the results in this work and provide a measure of what magnitude of estimate errors are tolerable in carrier landing scenarios.

REFERENCES

[1] D.-W. Gu, W. Kamal, and I. Postlethwaite, "A UAV Waypoint Generator," in *AIAA 1st Intelligent Systems Technical Conference*. American Institute of Aeronautics and Astronautics. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2004-6227

[2] I. L. Turner, M. D. Harley, and C. D. Drummond, "UAVs for coastal surveying," *Coastal Engineering*, vol. 114, pp. 19–24, Aug. 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378383916300370

[3] L. Geng, Y. F. Zhang, J. J. Wang, J. Y. H. Fuh, and S. H. Teo, "Mission planning of autonomous UAVs for urban surveillance with evolutionary algorithms," in *2013 10th IEEE International Conference on Control and Automation (ICCA)*, Jun. 2013, pp. 828–833.

[4] D. Pebrianti, F. Kendoul, S. Azrad, W. Wang, and K. Nonami, "Autonomous Hovering and Landing of a Quad-rotor Micro Aerial Vehicle by Means of on Ground Stereo Vision System," *Journal of System Design and Dynamics*, vol. 4, no. 2, pp. 269–284, 2010.

[5] W. Kong, D. Zhou, Y. Zhang, D. Zhang, X. Wang, B. Zhao, C. Yan, L. Shen, and J. Zhang, "A ground-based optical system for autonomous landing of a fixed wing UAV," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 4797–4804.

[6] J. T. Isaacs, K. O. Ezal, and J. P. Hespanha, "Local carrier-based precision approach and landing system," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. Las Vegas, NV, USA: IEEE, Dec. 2016, pp. 6284–6290. [Online]. Available: http://ieeexplore.ieee.org/document/7799236/

[7] R. Polvara, S. Sharma, J. Wan, A. Manning, and R. Sutton, "Towards autonomous landing on a moving vessel through fiducial markers," in *2017 European Conference on Mobile Robots (ECMR)*, Sep. 2017, pp. 1–6.

[8] C. Yu, J. Cai, and Q. Chen, "Multi-resolution visual fiducial and assistant navigation system for unmanned aerial vehicle landing," *Aerospace Science and Technology*, vol. 67, pp. 249–256, Aug. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1270963816312858

[9] O. A. Yakimenko, I. I. Kaminer, W. J. Lentz, and P. A. Ghyzel, "Unmanned aircraft navigation for shipboard landing using infrared vision," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 4, pp. 1181–1200, Oct. 2002.

[10] J. Shang and Z. Shi, *Vision-based Runway Recognition for UAV Autonomous Landing*.

[11] X. Pan, D. Ma, L. Jin, and Z. Jiang, "Vision-Based Approach Angle and Height Estimation for UAV Landing," in *2008 Congress on Image and Signal Processing*, vol. 3, May 2008, pp. 801–805.

[12] M. Laiacker, K. Kondak, M. Schwarzbach, and T. Muskardin, "Vision aided automatic landing system for fixed wing UAV," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2013, pp. 2971–2976, iSSN: 2153-0866.

[13] V. m. Raja, "Vision based landing for unmanned aerial vehicle," in *2011 Aerospace Conference*, Mar. 2011, pp. 1–8, iSSN: 1095-323X.

[14] S. Arora, S. Jain, S. Scherer, S. Nuske, L. Chamberlain, and S. Singh, "Infrastructure-free shipdeck tracking for autonomous landing," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 323–330.

[15] X. Chen, S. K. Phang, M. Shan, and B. M. Chen, "System integration of a vision-guided UAV for autonomous landing on moving platform," in *2016 12th IEEE International Conference on Control and Automation (ICCA)*, Jun. 2016, pp. 761–766, iSSN: null.

[16] H. Wang, Z. Xiong, J. C. Derenick, C. Stathis, and I. Cherepinsky, "Semantics based safe landing area detection for an unmanned vehicle," US Patent US9 177 481B2, Nov., 2015. [Online]. Available: https://patents.google.com/patent/US9177481B2/en

[17] D. Maturana and S. Scherer, "3D Convolutional Neural Networks for landing zone detection from LiDAR," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 3471–3478, iSSN: 1050-4729.

[18] F. Amzajerdian, V. E. Roback, A. E. Bulyshev, P. F. Brewster, W. A. Carrion, D. F. Pierrottet, G. D. Hines, L. B. Petway, B. W. Barnes, and A. M. Noe, "Imaging flash LIDAR for safe landing on solar system bodies and spacecraft rendezvous and docking," in *Laser Radar Technology and Applications XX; and Atmospheric Propagation XII*, vol. 9465. International Society for Optics and Photonics, May 2015, p. 946502. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/9465/946502/Imaging-flash-LIDAR-for-safe-landing-on-solar-system-bodies/10.1117/12.2178410.short

[19] M. GARRATT, H. POTA, A. LAMBERT, S. ECKERSLEYMASLIN, and C. FARABET, "Visual Tracking and LIDAR Relative Positioning for Automated Launch and Recovery of an Unmanned Rotorcraft from Ships at Sea," Jun. 2009. [Online]. Available: https://www.ingentaconnect.com/content/asne/nej/2009/00000121/00000002/art00014

[20] C. Toth, D. A. Grejner-Brzezinska, and Y.-J. Lee, "Terrain-based navigation: Trajectory recovery from LiDAR data," in *2008 IEEE/ION Position, Location and Navigation Symposium*, May 2008, pp. 760–765, iSSN: 2153-3598.

[21] P. G. Savage, *Strapdown Analytics*. Maple Plain, MN: Strapdown Associates, Inc., 2000.

[22] T. Hgstrm and J. Ehlin, "Measuring of a landing platform of a ship," US Patent US8 457 813B2, Jun., 2013. [Online]. Available: https://patents.google.com/patent/US8457813B2/en

[23] M. Uijt de Haag, D. Venable, M. Smearcheck, J. Campbell, and M. Miller, "Flash-LADAR Inertial Navigator Aiding." IEEE, 2006, pp. 677–683. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1650661

[24] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft Theory and Practice*. Princeton, NJ: Princeton University Press, 2012.

[25] E. Corporation, "EN-300 Precision Fiber Optic Inertial Measurement/Navigation Unit," Jul. 2018. [Online]. Available: https://emcore.com/wp-content/uploads/2018/06/EMCORE-EN-300-1-3-5-IMU.pdf

APPENDICES

APPENDIX A

Error State Linearization

This appendix contains linearization derivations for the velocity and quaternion equations. The last section also displays the complete F matrix used in the linear system.

A.1 Velocity Linearization

This section shows the error state vector differential equation for the velocity equations using the perturbation method described in the introduction of Section 5.4. The UAV velocity uses this derivation directly, while the carrier velocity follows a similar derivation. The only difference is that the carrier does not have an accelerometer onboard to measure the carrier's acceleration. Thus, the $(\underline{\tilde{\nu}}^b - \underline{\hat{b}}_a - \delta\underline{b}_a - \underline{n}_a)$ term is replaced with $(\underline{\hat{\omega}}^c + \delta\underline{\omega}^c)$ which ends up simplifying the procedure to a degree.

Starting from equation 5.4.2

$$\dot{\underline{\hat{v}}}^n + \delta\dot{\underline{v}}^n = \left[I - (\underline{\theta}_b^n\times)\right]\hat{R}_b^n\left(\underline{\tilde{\nu}}^b - \underline{\hat{b}}_a - \delta\underline{b}_a - \underline{n}_a\right) + \underline{g}^n \tag{A.0.1}$$

Distributing the first term yields

$$\dot{\underline{\hat{v}}}^n + \delta\dot{\underline{v}}^n = \hat{R}_b^n\left(\underline{\tilde{\nu}}^b - \underline{\hat{b}}_a - \delta\underline{b}_a - \underline{n}_a\right) - (\delta\underline{\theta}_b^n\times)\,\hat{R}_b^n\left(\underline{\tilde{\nu}}^b - \underline{\hat{b}}_a - \delta\underline{b}_a - \underline{n}_a\right) + \underline{g}^n \tag{A.0.2}$$

Subtracting the velocity component of equation 5.2.1

$$\delta\dot{\underline{v}}^n = \hat{R}_b^n\left(-\delta\underline{b}_a - \underline{n}_a\right) - (\delta\underline{\theta}_b^n\times)\,\hat{R}_b^n\left(\underline{\tilde{\nu}}^b - \underline{\hat{b}}_a - \delta\underline{b}_a - \underline{n}_a\right) \tag{A.0.3}$$

Reversing the cross product

$$\delta\dot{\underline{v}}^n = \hat{R}_b^n\left(-\delta\underline{b}_a - \underline{n}_a\right) + \left[\hat{R}_b^n\left(\underline{\tilde{\nu}}^b - \underline{\hat{b}}_a - \delta\underline{b}_a - \underline{n}_a\right)\right]\times\delta\underline{\theta}_b^n \tag{A.0.4}$$

Eliminating second order terms to produce a first-order approximation

$$\delta\underline{\dot{v}}^n = \hat{R}_b^n\left(-\delta\underline{b}_a - \underline{n}_a\right) + \left[\hat{R}_b^n\left(\underline{\tilde{\nu}}^b - \underline{\hat{b}}_a\right)\right] \times \delta\underline{\theta}_b^n \tag{A.0.5}$$

Rearranging to arrive at final result

$$\delta\underline{\dot{v}}^n = \left[\hat{R}_b^n\left(\underline{\tilde{\nu}}^b - \underline{\hat{b}}_a\right)\right] \times \delta\underline{\theta}_b^n - \hat{R}_b^n\delta\underline{b}_a - \hat{R}_b^n\underline{n}_a \tag{A.0.6}$$

A.2 Quaternion Linearization

This section shows the error state vector differential equation for the quaternion equations using the perturbation method described in the introduction of Section 5.4. The derivation below is specifically for the quaternion q_b^n, however, it applies to all quaternions in the state vector.

Recalling equation 5.1.3

$$\frac{d}{dt}\left(q_b^n\right) = \frac{1}{2}q_b^n \otimes \begin{bmatrix} 0 \\ \underline{\omega}^b \end{bmatrix} \tag{A.0.7}$$

where $q_b^n = \begin{bmatrix} 1 \\ -\frac{1}{2}\delta\underline{\theta}_b^n \end{bmatrix} \otimes \hat{q}_b^n$

Substituting this value into equation 4.3.2 produces

$$\frac{d}{dt}\left(\begin{bmatrix} 1 \\ -\frac{1}{2}\delta\underline{\theta}_b^n \end{bmatrix} \otimes \hat{q}_b^n\right) = \frac{1}{2}\begin{bmatrix} 1 \\ -\frac{1}{2}\delta\underline{\theta}_b^n \end{bmatrix} \otimes \hat{q}_b^n \otimes \begin{bmatrix} 0 \\ \underline{\omega}^b \end{bmatrix} \tag{A.0.8}$$

Expanding $\underline{\omega}^b$ and adding the perturbation results in 5.4.3

$$\frac{d}{dt}\left(\begin{bmatrix}1\\ -\frac{1}{2}\delta\underline{\theta}_b^n\end{bmatrix}\otimes\hat{q}_b^n\right)=\frac{1}{2}\begin{bmatrix}1\\ -\frac{1}{2}\delta\underline{\theta}_b^n\end{bmatrix}\otimes\hat{q}_b^n\otimes\begin{bmatrix}0\\ \tilde{\underline{\omega}}^b-\hat{\underline{b}}_g-\delta\underline{b}_g-\underline{n}_g\end{bmatrix} \tag{A.0.9}$$

To simplify notation, let $\delta q \triangleq \begin{bmatrix}1\\ -\frac{1}{2}\delta\underline{\theta}_b^n\end{bmatrix}$

$$\frac{d}{dt}\left(\delta q\otimes\hat{q}_b^n\right)=\frac{1}{2}\delta q\otimes\hat{q}_b^n\otimes\begin{bmatrix}0\\ \tilde{\underline{\omega}}^b-\hat{\underline{b}}_g-\delta\underline{b}_g-\underline{n}_g\end{bmatrix} \tag{A.0.10}$$

Applying the product rule on the left-hand side yields

$$\delta\dot{q}\otimes\hat{q}_b^n+\delta q\otimes\dot{\hat{q}}_b^n=\frac{1}{2}\delta q\otimes\hat{q}_b^n\otimes\begin{bmatrix}0\\ \tilde{\underline{\omega}}^b-\hat{\underline{b}}_g-\delta\underline{b}_g-\underline{n}_g\end{bmatrix} \tag{A.0.11}$$

Splitting the quaternion on the right into two parts

$$\delta\dot{q}\otimes\hat{q}_b^n+\delta q\otimes\dot{\hat{q}}_b^n=\frac{1}{2}\delta q\otimes\hat{q}_b^n\otimes\left(\begin{bmatrix}0\\ \tilde{\underline{\omega}}^b-\hat{\underline{b}}_g\end{bmatrix}+\begin{bmatrix}0\\ -\delta\underline{b}_g-\underline{n}_g\end{bmatrix}\right) \tag{A.0.12}$$

Distributing the quaternion-multiply operator on the right-hand side

$$\delta\dot{q}\otimes\hat{q}_b^n+\delta q\otimes\dot{\hat{q}}_b^n=\frac{1}{2}\delta q\otimes\hat{q}_b^n\otimes\begin{bmatrix}0\\ \tilde{\underline{\omega}}^b-\hat{\underline{b}}_g\end{bmatrix}+\frac{1}{2}\delta q\otimes\hat{q}_b^n\otimes\begin{bmatrix}0\\ -\delta\underline{b}_g-\underline{n}_g\end{bmatrix} \tag{A.0.13}$$

Substituting in the attitude component of equation 5.2.1

$$\delta\dot{q}\otimes\hat{q}_b^n+\delta q\otimes\dot{\hat{q}}_b^n=\delta q\otimes\dot{\hat{q}}_b^n+\frac{1}{2}\delta q\otimes\hat{q}_b^n\otimes\begin{bmatrix}0\\ -\delta\underline{b}_g-\underline{n}_g\end{bmatrix} \tag{A.0.14}$$

Canceling out common terms on both sides of equation

$$\delta\dot{q} \otimes \hat{q}_b^n = \frac{1}{2}\delta q \otimes \hat{q}_b^n \otimes \begin{bmatrix} 0 \\ -\delta\underline{b}_g - \underline{n}_g \end{bmatrix} \tag{A.0.15}$$

Right multiplying both sides of the equation by $(\hat{q}_b^n)^*$ yields

$$\delta\dot{q} = \frac{1}{2}\delta q \otimes \hat{q}_b^n \otimes \begin{bmatrix} 0 \\ -\delta\underline{b}_g - \underline{n}_g \end{bmatrix} \otimes (\hat{q}_b^n)^* \tag{A.0.16}$$

The last three terms on the right-hand side can equivalently be expressed with rotation matrices as follows

$$\delta\dot{q} = \frac{1}{2}\delta q \otimes \begin{bmatrix} 0 \\ -\hat{R}_b^n\delta\underline{b}_g - \hat{R}_b^n\underline{n}_g \end{bmatrix} \tag{A.0.17}$$

Substituting in the definition for δq

$$\begin{bmatrix} 1 \\ -\frac{1}{2}\delta\dot{\underline{\theta}}_b^n \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 \\ -\frac{1}{2}\delta\underline{\theta}_b^n \end{bmatrix} \otimes \begin{bmatrix} 0 \\ -\hat{R}_b^n\delta\underline{b}_g - \hat{R}_b^n\underline{n}_g \end{bmatrix} \tag{A.0.18}$$

Performing the quaternion multiplication yields

$$\begin{bmatrix} 1 \\ -\frac{1}{2}\delta\dot{\underline{\theta}}_b^n \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -\frac{1}{2}\delta\underline{\theta}_b^n\left(-\hat{R}_b^n\delta\underline{b}_g - \hat{R}_b^n\underline{n}_g\right) \\ -\frac{1}{2}\delta\underline{\theta}_b^n \times \left(-\hat{R}_b^n\delta\underline{b}_g - \hat{R}_b^n\underline{n}_g\right) - \hat{R}_b^n\delta\underline{b}_g - \hat{R}_b^n\underline{n}_g \end{bmatrix} \tag{A.0.19}$$

Factoring out the $\frac{1}{2}$ and canceling on both sides of the equation

$$\begin{bmatrix} 2 \\ -\delta\dot{\underline{\theta}}_b^n \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}\delta\underline{\theta}_b^n\left(-\hat{R}_b^n\delta\underline{b}_g - \hat{R}_b^n\underline{n}_g\right) \\ -\frac{1}{2}\delta\underline{\theta}_b^n \times \left(-\hat{R}_b^n\delta\underline{b}_g - \hat{R}_b^n\underline{n}_g\right) - \hat{R}_b^n\delta\underline{b}_g - \hat{R}_b^n\underline{n}_g \end{bmatrix} \tag{A.0.20}$$

Equating the second component of the matrices

$$-\delta\dot{\underline{\theta}}_b^n = -\frac{1}{2}\delta\underline{\theta}_b^n \times \left(-\hat{R}_b^n\delta\underline{b}_g - \hat{R}_b^n\underline{n}_g\right) - \hat{R}_b^n\delta\underline{b}_g - \hat{R}_b^n\underline{n}_g \tag{A.0.21}$$

Canceling out the second order terms to arrive at the final result

$$\delta\dot{\underline{\theta}}_b^n = \hat{R}_b^n \delta\underline{b}_g + \hat{R}_b^n \underline{n}_g \tag{A.0.22}$$

A.3 Linearized F Matrix

The complete F matrix as discussed in Section 5.4.

$$F = \begin{bmatrix}
0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} \\
0_{3\times3} & 0_{3\times3} & C & -\hat{R}_b^n & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} \\
0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \hat{R}_b^n & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} \\
0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -\frac{1}{\tau_a} I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} \\
0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -\frac{1}{\tau_g} I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} \\
0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} \\
0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} \\
0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} \\
0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} \\
0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & R1 & 0_{3\times1} & R2 & R3 \\
0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0 & 1 & 0 & 0 \\
0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & -\omega_n^2 & -2\zeta\omega_n & 0 & 0 \\
0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0 & 0 & -\frac{1}{\tau_{\omega y}} & 0 \\
0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0 & 0 & 0 & -\frac{1}{\tau_{\omega z}}
\end{bmatrix}$$

where C is the cross product matrix defined as $C \triangleq \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}$ and $\begin{bmatrix} x \\ y \\ z \end{bmatrix} \triangleq \hat{R}_b^n \left(\tilde{\underline{\nu}}^b - \hat{\underline{b}}_a \right)$

The elements $R1$, $R2$, and $R3$ are the first, second, and third columns of the rotation matrix from the carrier to the navigation frame, R_c^n.

A.4 Linearization Validation

The linear system described above was validated in simulation. This was accomplished by injecting errors into the system and propagating both the nonlinear equations and the corresponding linear equations. The time constants for the first order Markov equations were set to 1/10 of the simulation time to allow the equations to vary over a single Kalman cycle. Stopping the simulation after this cycle, the values of $\delta x_{nonlinear}$ and δx_{linear} were recorded and can be seen in Table A.1 with the last column displaying the error between them.

As seen from the table, the difference between the two sets of equations are negligible for most of the states. Since the carrier truth states are read in from a file instead of being propagated with truth equations, their errors are more difficult to analyze and determine if the linear approximation is accurate enough for the purpose of this study. These discrepancies are further evaluated in Chapter 6.

Table A.1: Linear system validation

Variable Name	δx_{linear}	$\delta x_{nonlinear}$	Error
$\delta \underline{p}^n$	1.2	1.1993	0.00070429
	2.1521	2.1506	0.0015336
	3.2949	3.2962	-0.0012571
$\delta \underline{v}^n$	0.29994	0.29853	0.0014106
	0.10557	0.10254	0.0030317
	0.29222	0.29471	-0.0024913
$\delta \underline{\theta}_b^n$	0.0099997	0.0099979	1.7492e-06
	0.020001	0.019998	3.4867e-06
	0.030001	0.029996	5.2574e-06
$\delta \underline{b}_a$	4.4538e-07	4.4538e-07	0
	8.9075e-07	8.9075e-07	0
	1.3361e-06	1.3361e-06	0
$\delta \underline{b}_g$	2.2011e-10	2.2011e-10	0
	4.4021e-10	4.4021e-10	0
	6.6032e-10	6.6032e-10	0
$\delta \underline{\theta}_k^b$	0.01	0.0099983	1.7495e-06
	0.02	0.019997	3.4991e-06
	0.03	0.029995	5.2486e-06
$\delta \underline{\theta}_l^b$	0.01	0.0099983	1.7495e-06
	0.02	0.019997	3.4991e-06
	0.03	0.029995	5.2486e-06
$\delta \underline{p}_c^n$	1.1	1.1027	-0.0026836
	2.2	2.2414	-0.041369
	3.3	3.3079	-0.0078574
$\delta \underline{v}_c^n$	0.1	0.10598	-0.0059808
	0.2	0.28102	-0.081023
	0.3	0.30876	-0.0087553
$\delta \underline{\theta}_c^n$	-0.041466	-0.043795	0.0023284
	0.014839	0.015533	-0.00069326
	0.029299	0.028934	0.00036456
$\delta \omega_{cx/n}^c$	0.049948	0.049693	0.00025503
$\delta \alpha_{cx/n}^c$	-0.0049471	-0.0055738	0.00062662
$\delta \omega_{cy/n}^c$	1.5848e-06	-0.00035573	0.00035732
$\delta \omega_{cz/n}^c$	7.9239e-07	0.00061737	-0.00061658

APPENDIX B

Measurement Model Linearization

This appendix contains linearization derivations for the the measurement models discussed in Section 5.5.

B.1 Line-of-Sight Linearization

This section linearizes the LOS measurement produced from the camera mounted on the UAV. The derivation used both Jacobians and perturbations to arrive at the final result.

Starting from

$$\tilde{\underline{z}}_k = \begin{bmatrix} \hat{l}_x^k / \hat{l}_z^k \\ \hat{l}_y^k / \hat{l}_z^k \end{bmatrix} \tag{B.0.1}$$

Defining intermediate function

$$h_{los}\left(\underline{l}^k\right) = \begin{bmatrix} l_x^k / l_z^k \\ l_y^k / l_z^k \end{bmatrix} \tag{B.0.2}$$

Defining the nominal

$$\hat{\underline{\tilde{z}}}_k \triangleq h_{los}\left(\hat{\underline{l}}^k\right) \tag{B.0.3}$$

Defining the perturbations

$$\tilde{\underline{z}}_k \triangleq \hat{\underline{\tilde{z}}}_k + \delta \underline{z}_k \tag{B.0.4}$$

$$\underline{l}^k \triangleq \hat{\underline{l}}^k + \delta \underline{l}^k \tag{B.0.5}$$

Substituting perturbations into original equation yields

$$\hat{\underline{\tilde{z}}}_k + \delta \underline{z}_k = h_{los}\left(\underline{l}^k + \delta \underline{l}^k\right) \tag{B.0.6}$$

Performing a first-order Taylor Series expansion

$$\hat{\underline{z}}_k + \delta \underline{z}_k \approx h_{los}\left(\hat{\underline{l}}^k\right) + \left(\left.\frac{\partial h_{los}}{\partial l^k}\right|_{l^k=\hat{l}^k}\right)\delta \underline{l}^k \tag{B.0.7}$$

Subtracting the nominal

$$\delta \underline{z}_k = \left(\left.\frac{\partial h_{los}}{\partial l^k}\right|_{l^k=\hat{l}^k}\right)\delta \underline{l}^k \tag{B.0.8}$$

Performing the jacobian and letting it equal H_{los}

$$H_{los} \triangleq \begin{bmatrix} \frac{1}{l_z^k} & 0 & -\frac{l_x^k}{\left(l_z^k\right)^2} \\ 0 & \frac{1}{l_z^k} & -\frac{l_y^k}{\left(l_z^k\right)^2} \end{bmatrix} \tag{B.0.9}$$

Substituting in H_{los}

$$\delta \underline{z}_k = H_{los}\delta \underline{l}^k \tag{B.0.10}$$

Now linearizing $\delta \underline{l}^k$ starting from

$$\underline{l}^k = R_b^k R_n^b\left(\underline{p}_c + R_c^n \underline{r}_f^c - \underline{p}^n\right) \tag{B.0.11}$$

Defining the nominal

$$\hat{\underline{l}}^k = \hat{R}_b^k \hat{R}_n^b\left(\hat{\underline{p}}_c + \hat{R}_c^n \underline{r}_f^c - \hat{\underline{p}}^n\right) \tag{B.0.12}$$

Defining the perturbations

$$\underline{l}^k = \hat{\underline{l}}^k + \delta \underline{l}^k \tag{B.0.13}$$

$$\underline{p}^n = \hat{\underline{p}}^n + \delta \underline{p}^n \tag{B.0.14}$$

$$\underline{p}_c^n = \hat{\underline{p}}_c^n + \delta \underline{p}_c^n \tag{B.0.15}$$

$$R_k^b \triangleq \left[I - \left(\delta\theta_k^b\times\right)\right]\hat{R}_k^b \implies R_b^k \triangleq \hat{R}_b^k\left[I + \left(\delta\theta_k^b\times\right)\right] \tag{B.0.16}$$

$$R_b^n \triangleq \left[I - \left(\delta\theta_b^n\times\right)\right]\hat{R}_b^n \implies R_n^b \triangleq \hat{R}_n^b\left[I + \left(\delta\theta_b^n\times\right)\right] \tag{B.0.17}$$

$$R_c^n \triangleq \left[I - \left(\delta\theta_c^n\times\right)\right]\hat{R}_c^n \tag{B.0.18}$$

Substituting perturbations into original $\delta \underline{l}^k$ equation yields

$$\hat{\underline{l}}^k + \delta \underline{l}^k = \hat{R}_b^k \left[I + \left(\delta\theta_k^b \times \right) \right] \hat{R}_n^b \left[I + \left(\delta\theta_b^n \times \right) \right] \left(\hat{\underline{p}}_c^n + \delta \underline{p}_c^n + \left[I - \left(\delta\theta_c^n \times \right) \right] \hat{R}_c^n \underline{r}_f^c - \hat{\underline{p}}^n - \delta \underline{p}^n \right) \tag{B.0.19}$$

Expanding

$$\hat{\underline{l}}^k + \delta \underline{l}^k = \left[\hat{R}_b^k + \hat{R}_b^k \left(\delta\theta_k^b \times \right) \right] \left[\hat{R}_n^b + \hat{R}_n^b \left(\delta\theta_b^n \times \right) \right] \left(\hat{\underline{p}}_c^n + \delta \underline{p}_c^n + \left[\hat{R}_c^n - \left(\delta\theta_c^n \times \right) \hat{R}_c^n \right] \underline{r}_f^c - \hat{\underline{p}}^n - \delta \underline{p}^n \right) \tag{B.0.20}$$

Further expanding while removing higher order terms

$$\begin{aligned} \hat{\underline{l}}^k + \delta \underline{l}^k = & \hat{R}_b^k \hat{R}_n^b \hat{\underline{p}}_c^n + \hat{R}_b^k \hat{R}_n^b \delta \underline{p}_c^n + \hat{R}_b^k \hat{R}_n^b \hat{R}_c^n \underline{r}_f^c - \hat{R}_b^k \hat{R}_n^b \left(\delta\theta_c^n \times \right) \hat{R}_c^n \underline{r}_f^c - \hat{R}_b^k \hat{R}_n^b \hat{\underline{p}}^n \\ & - \hat{R}_b^k \hat{R}_n^b \delta \underline{p}^n + \hat{R}_b^k \hat{R}_n^b \left(\delta\theta_b^n \times \right) \hat{\underline{p}}_c^n - \hat{R}_b^k \hat{R}_n^b \left(\delta\theta_b^n \times \right) \hat{R}_c^n \underline{r}_f^c - \hat{R}_b^k \hat{R}_n^b \left(\delta\theta_b^n \times \right) \hat{\underline{p}}^n \\ & + \hat{R}_b^k \left(\delta\theta_k^b \times \right) \hat{R}_n^b \hat{\underline{p}}_c^n + \hat{R}_b^k \left(\delta\theta_k^b \times \right) \hat{R}_n^b \hat{R}_c^n \underline{r}_f^c - \hat{R}_b^k \left(\delta\theta_k^b \times \right) \hat{R}_n^b \hat{\underline{p}}^n \end{aligned} \tag{B.0.21}$$

Canceling the nominal

$$\begin{aligned} \delta \underline{l}^k = & \hat{R}_b^k \hat{R}_n^b \delta \underline{p}_c^n - \hat{R}_b^k \hat{R}_n^b \left(\delta\theta_c^n \times \right) \hat{R}_c^n \underline{r}_f^c - \hat{R}_b^k \hat{R}_n^b \delta \underline{p}^n + \hat{R}_b^k \hat{R}_n^b \left(\delta\theta_b^n \times \right) \hat{\underline{p}}_c^n - \hat{R}_b^k \hat{R}_n^b \left(\delta\theta_b^n \times \right) \hat{R}_c^n \underline{r}_f^c \\ & - \hat{R}_b^k \hat{R}_n^b \left(\delta\theta_b^n \times \right) \hat{\underline{p}}^n + \hat{R}_b^k \left(\delta\theta_k^b \times \right) \hat{R}_n^b \hat{\underline{p}}_c^n + \hat{R}_b^k \left(\delta\theta_k^b \times \right) \hat{R}_n^b \hat{R}_c^n \underline{r}_f^c - \hat{R}_b^k \left(\delta\theta_k^b \times \right) \hat{R}_n^b \hat{\underline{p}}^n \end{aligned} \tag{B.0.22}$$

Making use of the cross product property $a \times b = -b \times a$ and rearranging so δx is on the right

$$\begin{aligned} \delta \underline{l}^k = & \left[\hat{R}_b^k \hat{R}_n^b \right] \delta \underline{p}_c^n - \left[\hat{R}_b^k \hat{R}_n^b \left(-\hat{R}_c^n \underline{r}_f^c \times \right) \right] \delta\theta_c^n - \left[\hat{R}_b^k \hat{R}_n^b \right] \delta \underline{p}^n \\ & + \left[\hat{R}_b^k \hat{R}_n^b \left(-\hat{\underline{p}}_c^n \times \right) \right] \delta\theta_b^n + \left[\hat{R}_b^k \hat{R}_n^b \left(-\hat{R}_c^n \underline{r}_f^c \times \right) \right] \delta\theta_b^n - \left[\hat{R}_b^k \hat{R}_n^b \left(-\hat{\underline{p}}^n \times \right) \right] \delta\theta_b^n \\ & + \left[\hat{R}_b^k \left(-\hat{R}_n^b \hat{\underline{p}}_c^n \times \right) \right] \delta\theta_k^b + \left[\hat{R}_b^k \left(-\hat{R}_n^b \hat{R}_c^n \underline{r}_f^c \times \right) \right] \delta\theta_k^b - \left[\hat{R}_b^k \left(-\hat{R}_n^b \hat{\underline{p}}^n \times \right) \right] \delta\theta_k^b \end{aligned} \tag{B.0.23}$$

Simplifying and grouping terms as products of δx

$$\begin{aligned}\delta \underline{l}^k = &\left[\hat{R}_b^k \hat{R}_n^b\right] \delta \underline{p}_{\underline{c}}^n - \left[\hat{R}_b^k \hat{R}_n^b \left(-\hat{R}_c^n \underline{r}_{\underline{f}}^c \times\right)\right] \delta\theta_c^n - \left[\hat{R}_b^k \hat{R}_n^b\right] \delta \underline{p}^n \\ &- \left[\hat{R}_b^k \hat{R}_n^b \left(\left(\hat{\underline{p}}_{\underline{c}}^n \times\right) + \left(\hat{R}_c^n \underline{r}_{\underline{f}}^c \times\right) - \left(\hat{\underline{p}}^n \times\right)\right)\right] \delta\theta_b^n \\ &- \left[\hat{R}_b^k \left(\left(\hat{R}_n^b \hat{\underline{p}}_{\underline{c}}^n \times\right) + \left(\hat{R}_n^b \hat{R}_c^n \underline{r}_{\underline{f}}^c \times\right) - \left(\hat{R}_n^b \hat{\underline{p}}^n \times\right)\right)\right] \delta\theta_k^b\end{aligned} \tag{B.0.24}$$

Letting the δx coefficients be grouped into matrix form and letting it equal H_l

$$\begin{aligned}H_l \triangleq \Big[&- \hat{R}_b^k \hat{R}_n^b \quad 0_{3\times3} \quad -\hat{R}_b^k \hat{R}_n^b \left(\left(\hat{\underline{p}}_{\underline{c}}^n \times\right) + \left(\hat{R}_c^n \underline{r}_{\underline{f}}^c \times\right) - \left(\hat{\underline{p}}^n \times\right)\right) \quad 0_{3\times3} \\ &0_{3\times3} \quad -\hat{R}_b^k \left(\left(\hat{R}_n^b \hat{\underline{p}}_{\underline{c}}^n \times\right) + \left(\hat{R}_n^b \hat{R}_c^n \underline{r}_{\underline{f}}^c \times\right) - \left(\hat{R}_n^b \hat{\underline{p}}^n \times\right)\right) \quad 0_{3\times3} \\ &\hat{R}_b^k \hat{R}_n^b \quad 0_{3\times3} \quad \hat{R}_b^k \hat{R}_n^b \left(\hat{R}_c^n \underline{r}_{\underline{f}}^c \times\right) \quad 0_{3\times1} \quad 0_{3\times1} \quad 0_{3\times1} \quad 0_{3\times1} \Big]\end{aligned} \tag{B.0.25}$$

Substituting H_l

$$\delta \underline{l}^k = H_l \delta \underline{x} \tag{B.0.26}$$

Recalling $\delta \underline{z}_k = H_{los} \delta \underline{l}^k$ and substituting $\delta \underline{l}^k$ as defined above

$$\delta \underline{z}_k = H_{los} H_l \delta \underline{x} \tag{B.0.27}$$

This leads to the final linearization in matrix form where $H_k \triangleq H_{los} H_l$

$$\delta \underline{z}_k = H_k \delta \underline{x} \tag{B.0.28}$$

B.2 Lidar Linearization

This section linearizes the lidar measurement produced from the lidar mounted on the UAV.

Starting from

$$\tilde{\underline{z}}_l = L_m R_b^l R_n^b R_c^n \underline{u}^c \tag{B.0.29}$$

where $\underline{u}^c = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^T$ and $L_m = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$

Defining the nominal

$$\hat{\underline{z}}_l \triangleq L_m \hat{R}_b^l \hat{R}_n^b \hat{R}_c^n \underline{u}^c \tag{B.0.30}$$

Defining the perturbations

$$\tilde{\underline{z}}_l \triangleq \hat{\underline{z}}_l + \delta \underline{z}_l \tag{B.0.31}$$

$$R_l^b \triangleq \left[I - \left(\delta\theta_l^b \times \right) \right] \hat{R}_l^b \implies R_b^l \triangleq \hat{R}_b^l \left[I + \left(\delta\theta_l^b \times \right) \right] \tag{B.0.32}$$

$$R_b^n \triangleq \left[I - \left(\delta\theta_b^n \times \right) \right] \hat{R}_b^n \implies R_n^b \triangleq \hat{R}_n^b \left[I + \left(\delta\theta_b^n \times \right) \right] \tag{B.0.33}$$

$$R_c^n \triangleq \left[I - \left(\delta\theta_c^n \times \right) \right] \hat{R}_c^n \tag{B.0.34}$$

Substituting perturbations into original equation yields

$$\hat{\underline{z}}_l + \delta \underline{z}_l = L_m \left(\hat{R}_b^l \left[I + \left(\delta\theta_l^b \times \right) \right] \hat{R}_n^b \left[I + \left(\delta\theta_b^n \times \right) \right] \left[I - \left(\delta\theta_c^n \times \right) \right] \right) \hat{R}_c^n \underline{u}^c \tag{B.0.35}$$

Expanding

$$\hat{\underline{z}}_l + \delta \underline{z}_l = L_m \left(\left[\hat{R}_b^l + \hat{R}_b^l \left(\delta\theta_l^b \times \right) \right] \left[\hat{R}_n^b + \hat{R}_n^b \left(\delta\theta_b^n \times \right) \right] \left[\hat{R}_c^n - \left(\delta\theta_c^n \times \right) \hat{R}_c^n \right] \right) \underline{u}^c \tag{B.0.36}$$

Further expanding while removing higher order terms

$$\hat{\underline{z}}_l + \delta \underline{z}_l = L_m \hat{R}_b^l \hat{R}_n^b \hat{R}_c^n \underline{u}^c - L_m \hat{R}_b^l \hat{R}_n^b \left(\delta\theta_c^n \times \right) \hat{R}_c^n \underline{u}^c + L_m \hat{R}_b^l \hat{R}_n^b \left(\delta\theta_b^n \times \right) \hat{R}_c^n \underline{u}^c + L_m \hat{R}_b^l \left(\delta\theta_l^b \times \right) \hat{R}_n^b \hat{R}_c^n \underline{u}^c \tag{B.0.37}$$

Canceling the nominal

$$\delta \underline{z}_l = -L_m \hat{R}_b^l \hat{R}_n^b \left(\delta\theta_c^n \times \right) \hat{R}_c^n \underline{u}^c + L_m \hat{R}_b^l \hat{R}_n^b \left(\delta\theta_b^n \times \right) \hat{R}_c^n \underline{u}^c + L_m \hat{R}_b^l \left(\delta\theta_l^b \times \right) \hat{R}_n^b \hat{R}_c^n \underline{u}^c \tag{B.0.38}$$

Making use of the cross product property $a \times b = -b \times a$ and rearranging so $\delta\theta$ is on the right

$$\delta\underline{z}_l = \left[-L_m\hat{R}_b^l\hat{R}_n^b\left(-\hat{R}_c^n\underline{u}^c\times\right)\right]\delta\theta_c^n + \left[L_m\hat{R}_b^l\hat{R}_n^b\left(-\hat{R}_c^n\underline{u}^c\times\right)\right]\delta\theta_b^n + \left[L_m\hat{R}_b^l\left(-\hat{R}_n^b\hat{R}_c^n\underline{u}^c\times\right)\right]\delta\theta_l^b \tag{B.0.39}$$

Simplifying to arrive at linear result

$$\delta\underline{z}_l = \left[L_m\hat{R}_b^l\hat{R}_n^b\left(\hat{R}_c^n\underline{u}^c\times\right)\right]\delta\theta_c^n - \left[L_m\hat{R}_b^l\hat{R}_n^b\left(\hat{R}_c^n\underline{u}^c\times\right)\right]\delta\theta_b^n - \left[L_m\hat{R}_b^l\left(\hat{R}_n^b\hat{R}_c^n\underline{u}^c\times\right)\right]\delta\theta_l^b \tag{B.0.40}$$

Letting the δx coefficients be grouped into matrix form and letting it equal H_l

$$H_l \triangleq \begin{bmatrix} 0_{2\times6} & -L_m\hat{R}_b^l\hat{R}_n^b\left(\hat{R}_c^n\underline{u}^c\times\right) & 0_{2\times9} & -L_m\hat{R}_b^l\left(\hat{R}_n^b\hat{R}_c^n\underline{u}^c\times\right) & 0_{2\times6} & L_m\hat{R}_b^l\hat{R}_n^b\left(\hat{R}_c^n\underline{u}^c\times\right) & 0_{2\times4} \end{bmatrix} \tag{B.0.41}$$

This leads to the final linearization in matrix form

$$\delta\underline{z}_l = H_l\delta\underline{x} \tag{B.0.42}$$

B.3 GPS Linearization

This section shows the linearization of the UAV GPS measurements. Since the measurement is already linear the error state vector differential equation is produced without much computation.

Starting from

$$\tilde{\underline{z}}_{gps} = \underline{p}^n \tag{B.0.43}$$

Defining the nominal

$$\hat{\tilde{\underline{z}}}_{gps} \triangleq \hat{\underline{p}}^n \tag{B.0.44}$$

Defining the perturbations

$$\tilde{\underline{z}}_{gps} \triangleq \hat{\tilde{\underline{z}}}_{gps} + \delta\underline{z}_{gps} \tag{B.0.45}$$

$$\underline{p}^n \triangleq \hat{\underline{p}}^n + \delta\underline{p}^n \tag{B.0.46}$$

Substituting perturbations into original equation yields

$$\hat{\tilde{\underline{z}}}_{gps} + \delta\underline{z}_{gps} = \hat{\underline{p}}^n + \delta\underline{p}^n \tag{B.0.47}$$

Canceling the nominal

$$\delta\underline{z}_{gps} = \delta\underline{p}^n \tag{B.0.48}$$

Defining H_{gps}

$$H_{gps} \triangleq \begin{bmatrix} I_{3\times3} & 0_{3\times31} \end{bmatrix} \tag{B.0.49}$$

Putting final linearization in matrix form to arrive at result

$$\delta\underline{z}_{gps} = H_{gps}\delta\underline{x} \tag{B.0.50}$$

B.4 Linearization Validation

As was done with the linearization of the design model in Section 5.4, the measurement linearization was validated in simulation to ensure that the approximation is accurate. After running the simulation over a single Kalman cycle and injecting errors into the state estimates using equation 4.3.3, the linear residuals were compared to the nonlinear residuals. The results are found in Table B.1.

Restating the nonlinear residuals which were defined in equations 5.3.11 - 5.3.13 as

$$\delta\underline{z}_{k,nonlinear} \triangleq \tilde{\underline{z}}_k - \hat{\tilde{\underline{z}}}_k \tag{B.0.51}$$

$$\delta\underline{z}_{l,nonlinear} \triangleq \tilde{\underline{z}}_l - \hat{\tilde{\underline{z}}}_l \tag{B.0.52}$$

$$\delta\underline{z}_{gps,nonlinear} \triangleq \tilde{\underline{z}}_{gps} - \hat{\tilde{\underline{z}}}_{gps} \tag{B.0.53}$$

Table B.1: Linear measurement validation

Measurement	$\delta \underline{z}_{linear}$	$\delta \underline{z}_{nonlinear}$	Error
LOS_x	0.013086	0.015454	-0.0023678
LOS_y	-0.012642	-0.01132	-0.0013219
Lidar_x	-0.0013904	-0.0013932	2.7988e-06
Lidar_y	0.00099393	0.00099097	2.9584e-06
GPS_x	1	1	0
GPS_y	2	2	0
GPS_z	3	3	0

with the linear residuals being defined in equation 5.5.1 while neglecting the noise term

$$\delta \underline{z}_{linear} \triangleq H \delta \underline{x} \tag{B.0.54}$$

which is an augmentation of all three measurements.

As seen in Table B.1, the errors in the LOS measurements are small, but may not be small enough. This is likely due to the necessity of estimating the position of the carrier which introduces additional rotations and lever arms for the vector sum. These discrepancies will be further evaluated in Section 6.6. As for the lidar measurements, the errors are orders of magnitude less than the residuals which signifies they are consistent with the nonlinear measurements. The errors in the GPS are zero since the measurement model was linear to begin with and thus the nonlinear and linear residuals are identical.

APPENDIX C

Filter Validation

This appendix includes additional EKF validation plots that were not included in Chapter 6 for brevity.

C.1 Nonlinear State Propagation

This section contains additional nonlinear propagation plots. See Section 6.2 for more information.

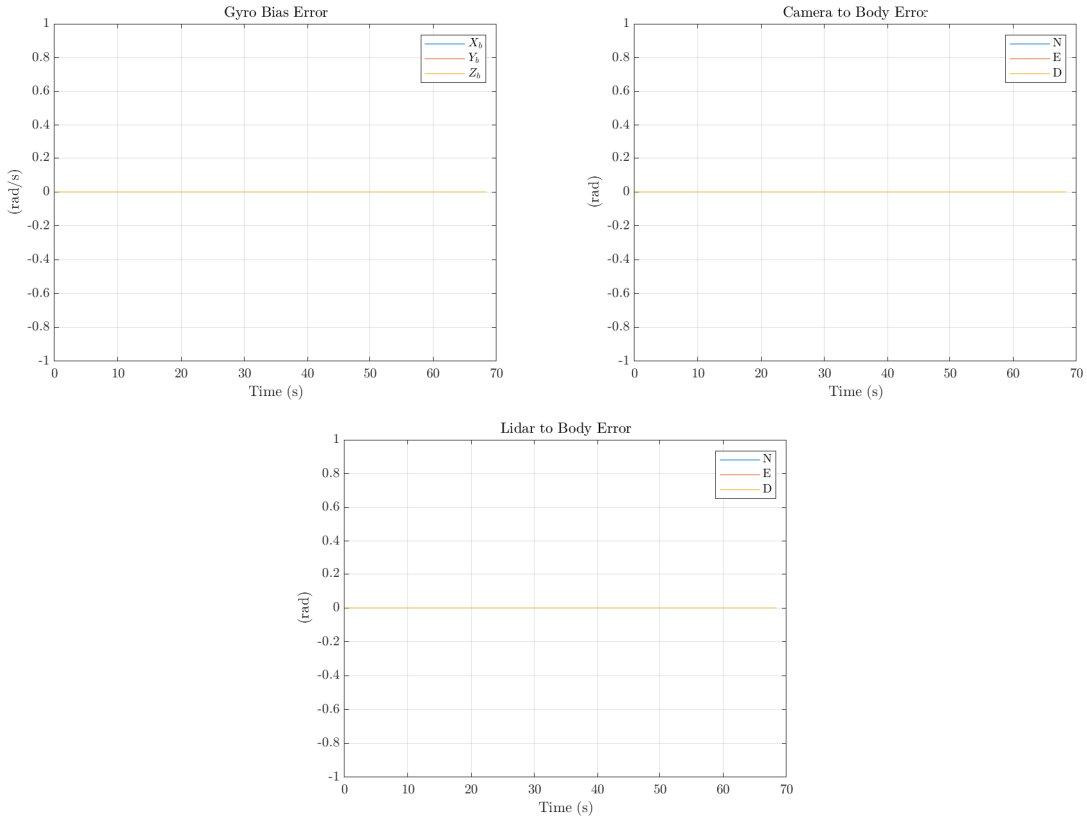

Fig. C.1: UAV parameter state errors

C.2 Covarinace Propagation

This section contains additional hairline covariance plots. See Section 6.5 for more information.

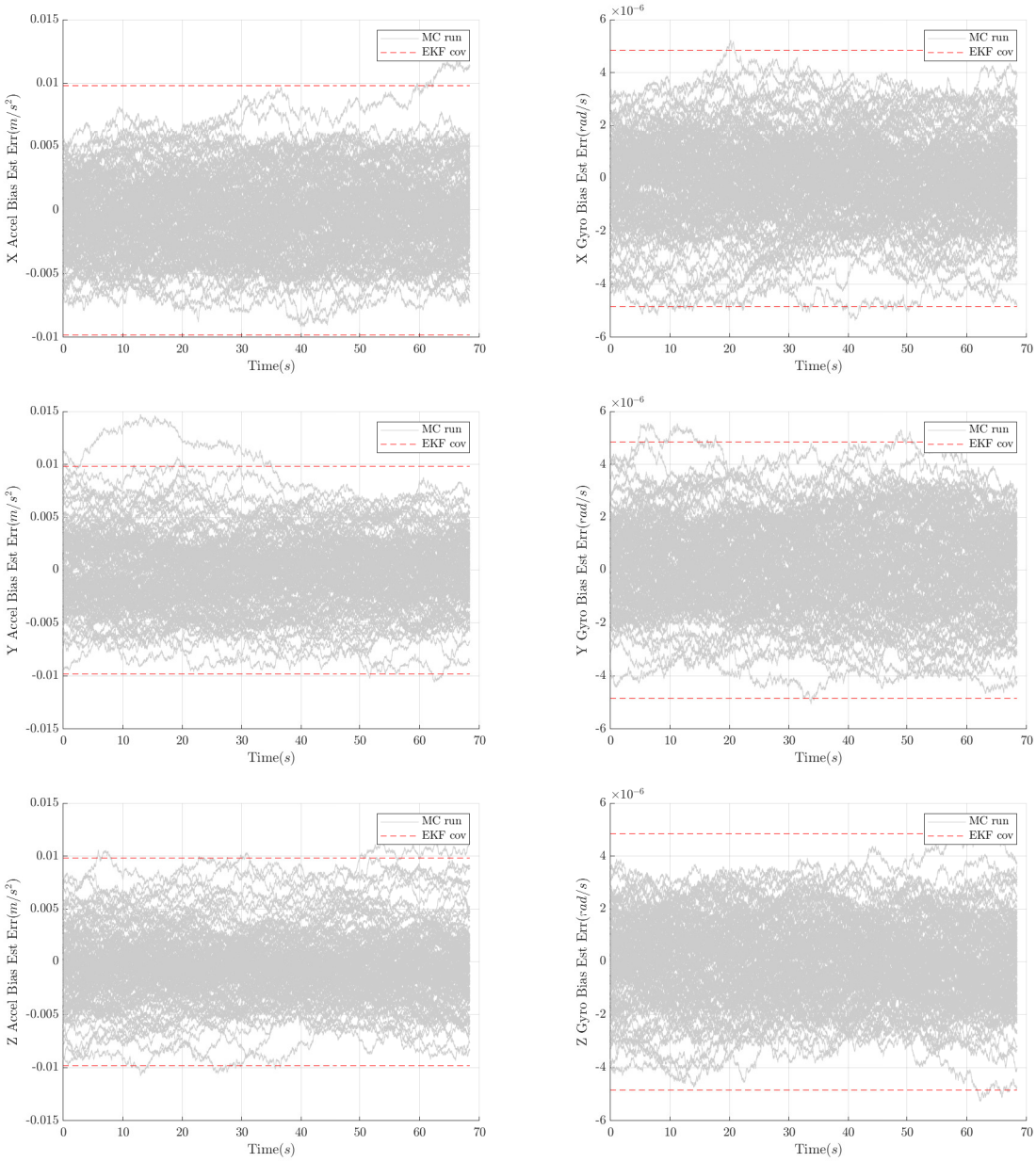

Fig. C.2: Accelerometer and gyroscope biases

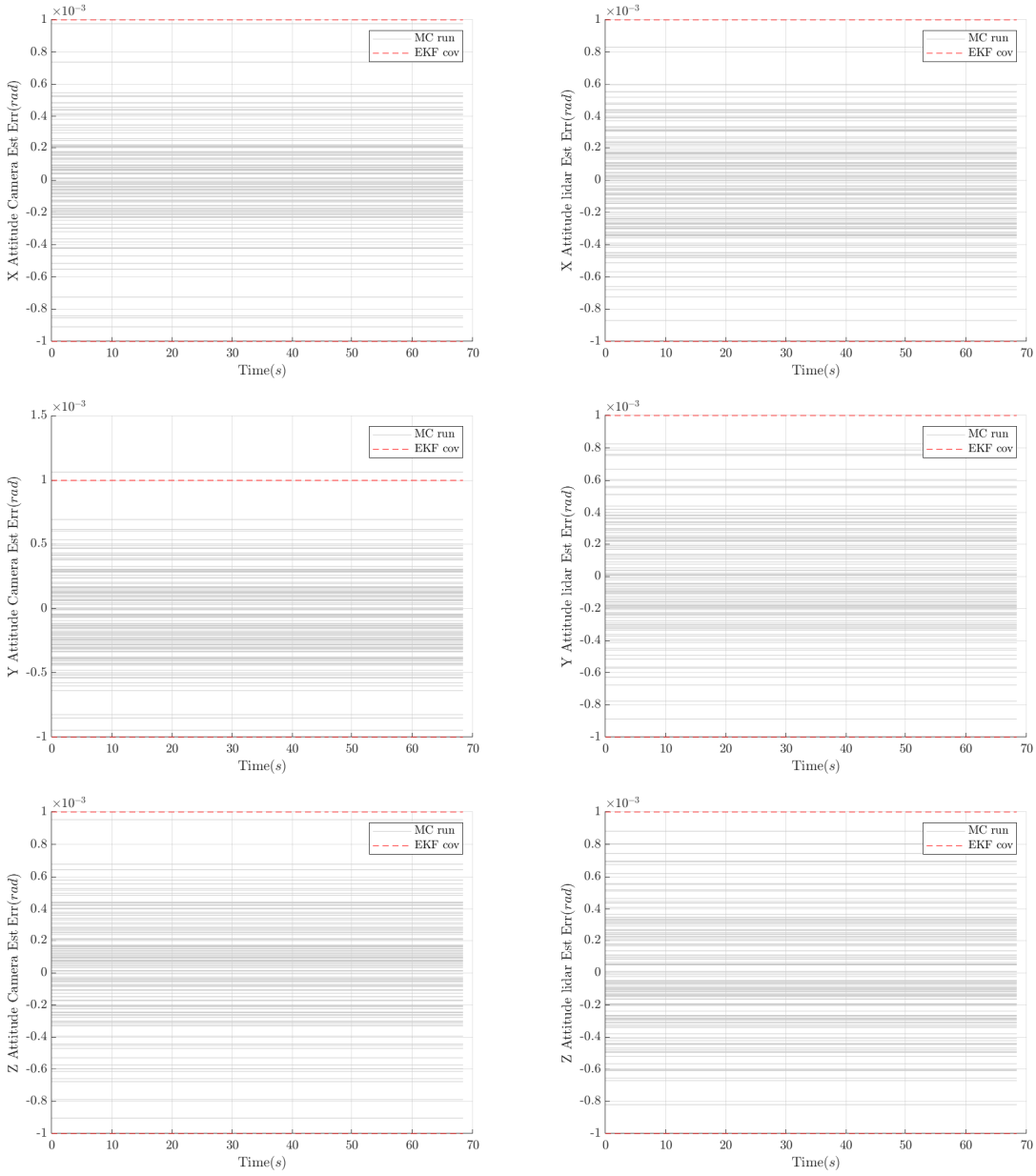

Fig. C.3: Camera and lidar misalignment

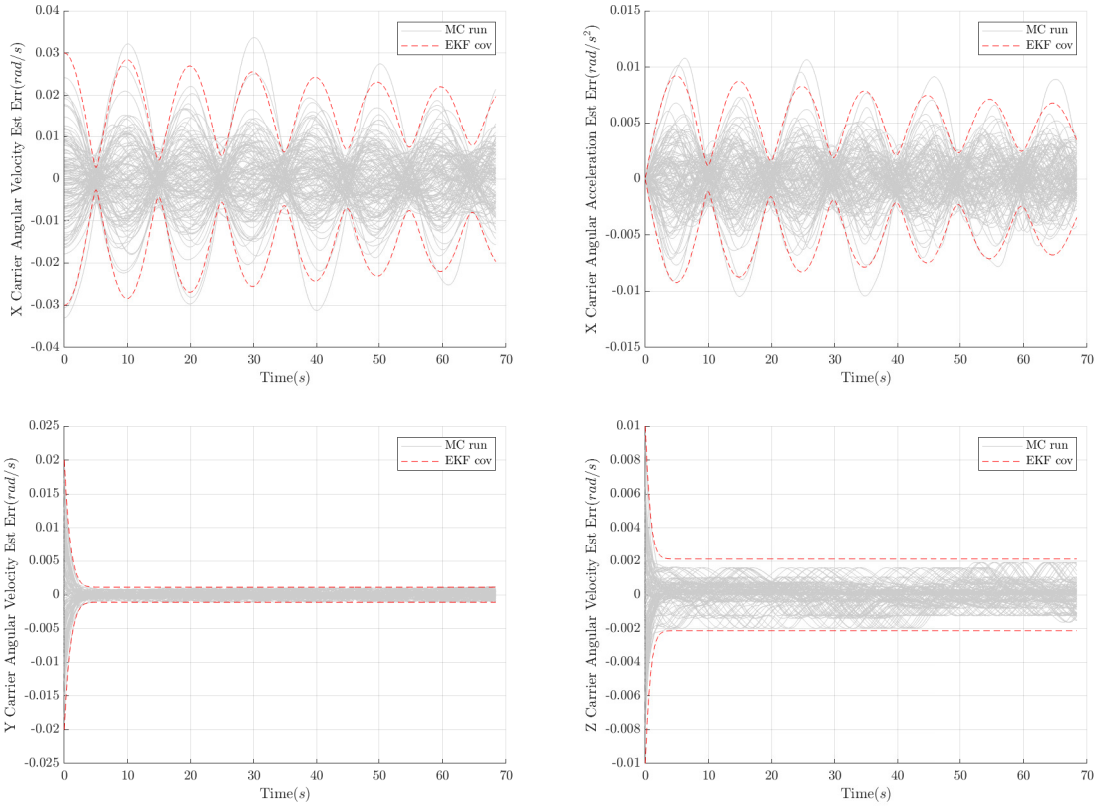

Fig. C.4: Carrier attitude rates

C.3 State Estimation

This section contains additional estimation hairline plots. See Section 6.6 for more information.

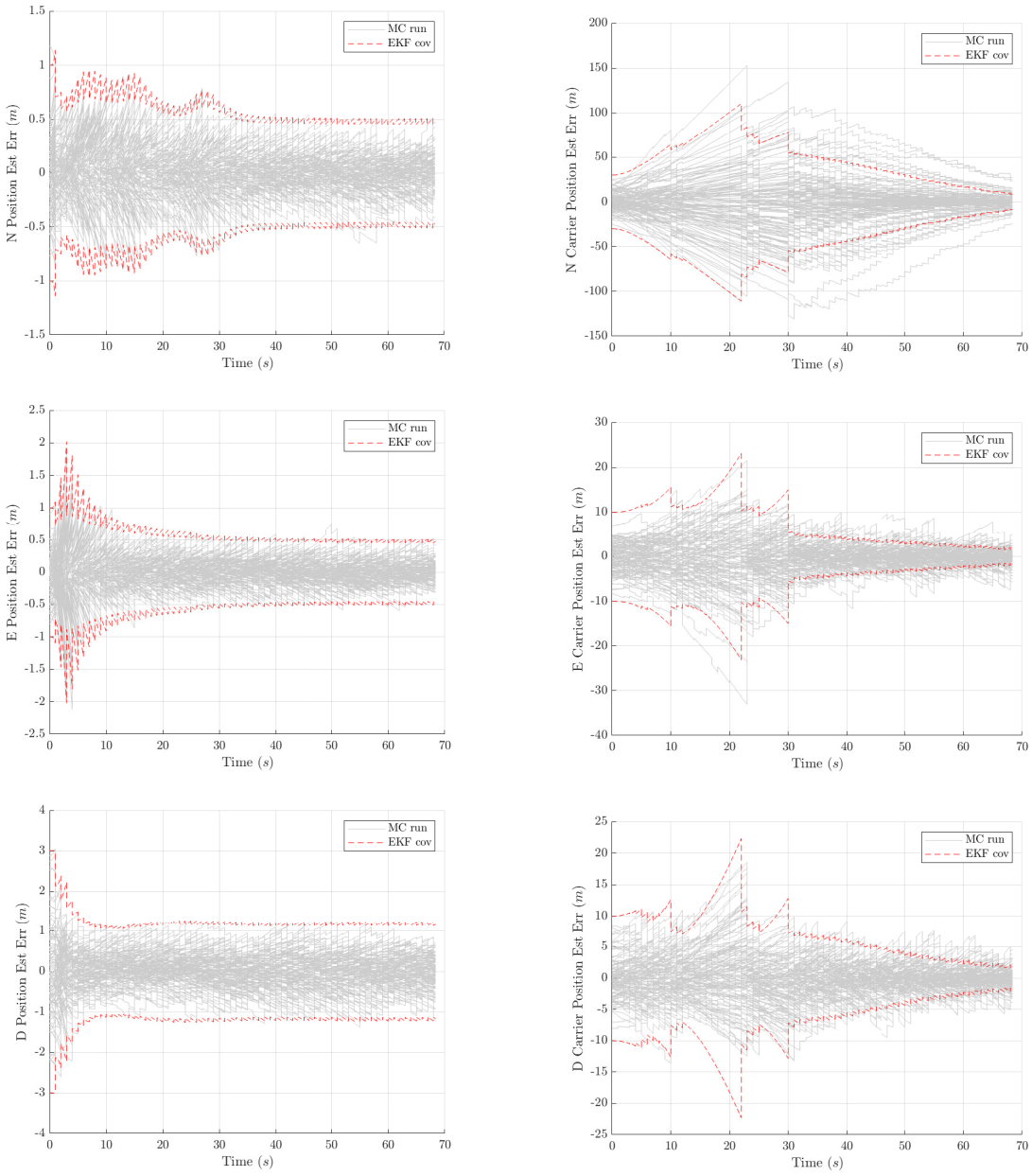

Fig. C.5: UAV and carrier positions estimate

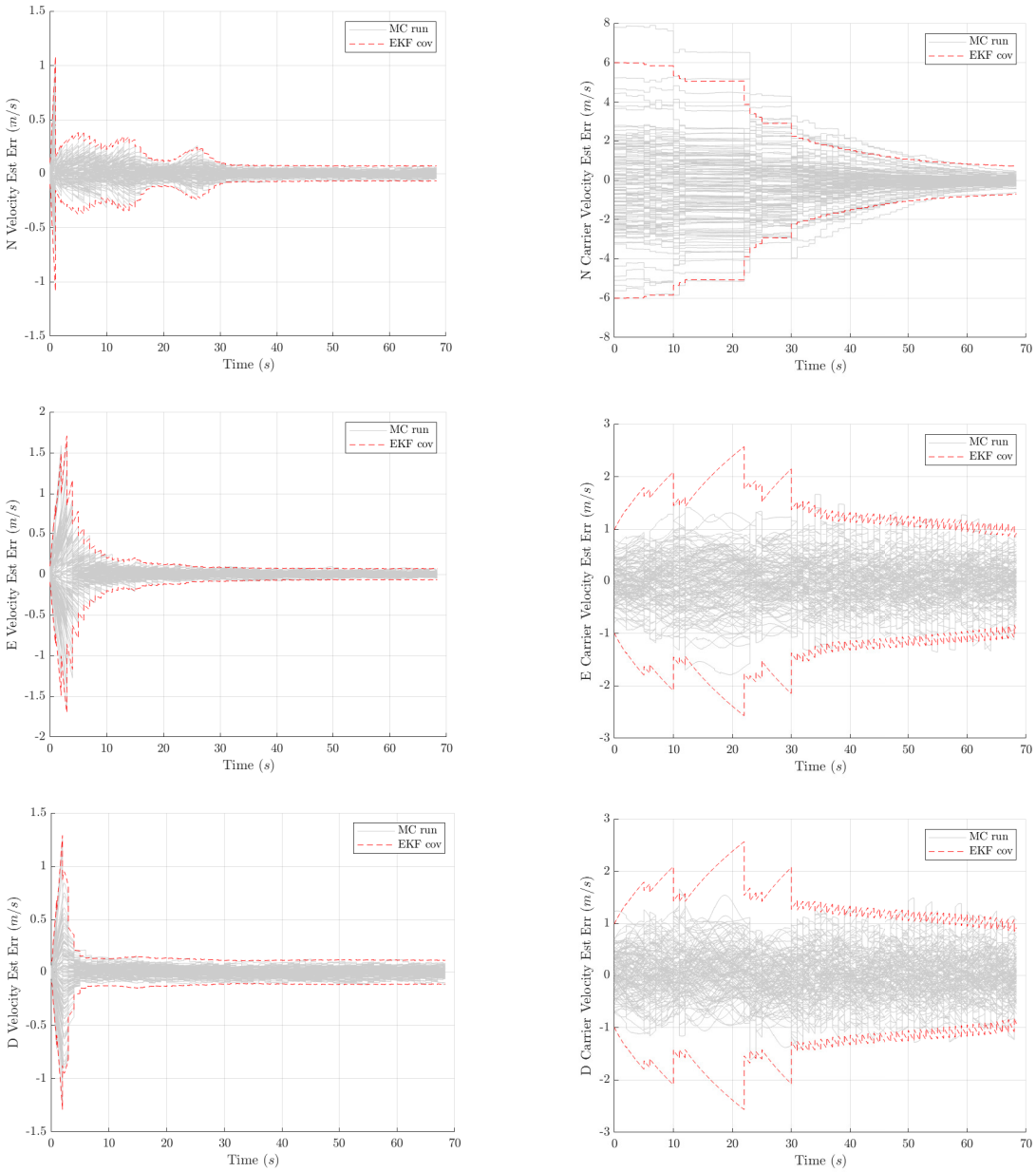

Fig. C.6: UAV and carrier velocities estimate

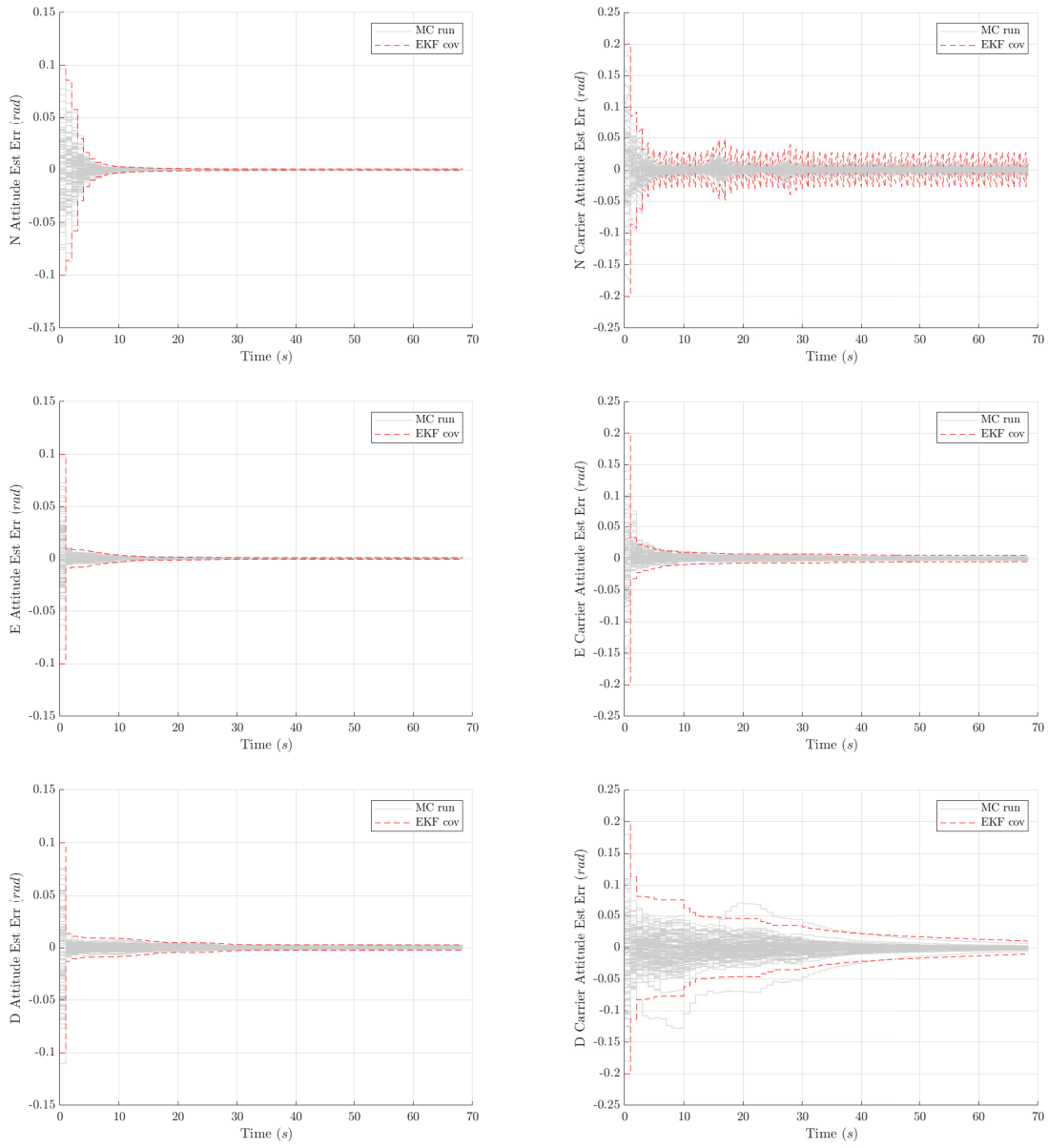

Fig. C.7: UAV and carrier attitudes estimate

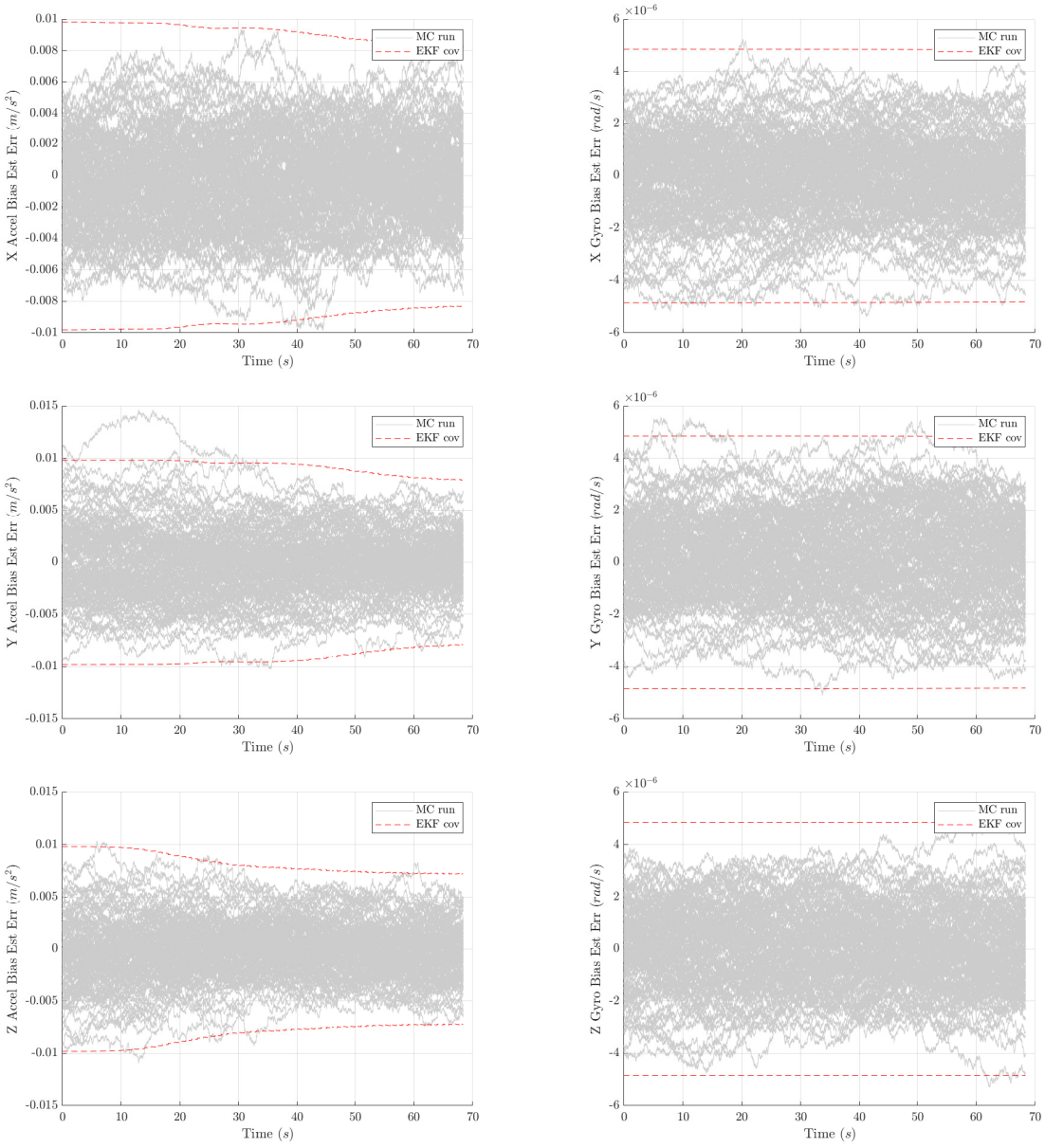

Fig. C.8: Accelerometer and gyro bias estimate

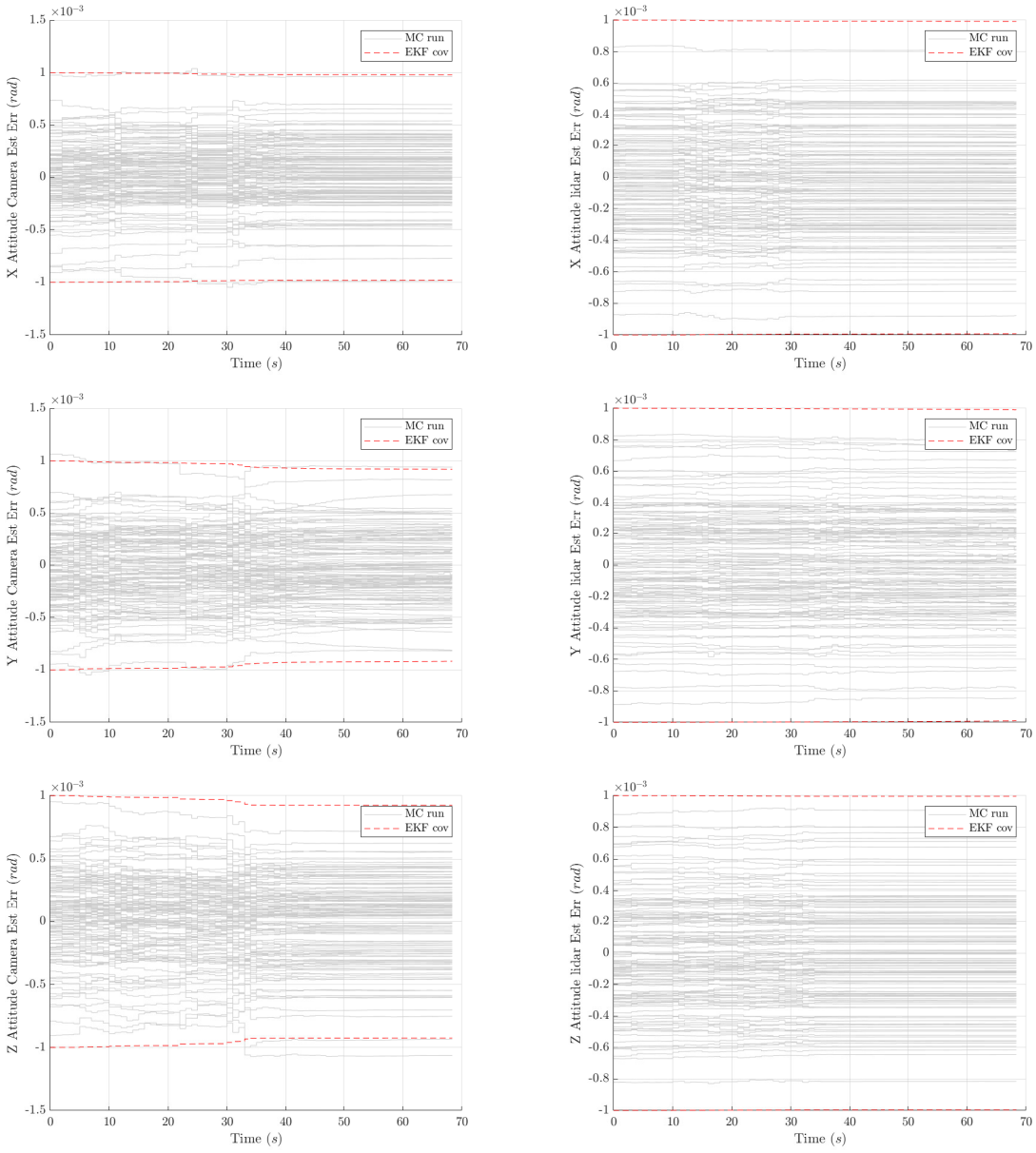

Fig. C.9: Camera and lidar misalignment estimate

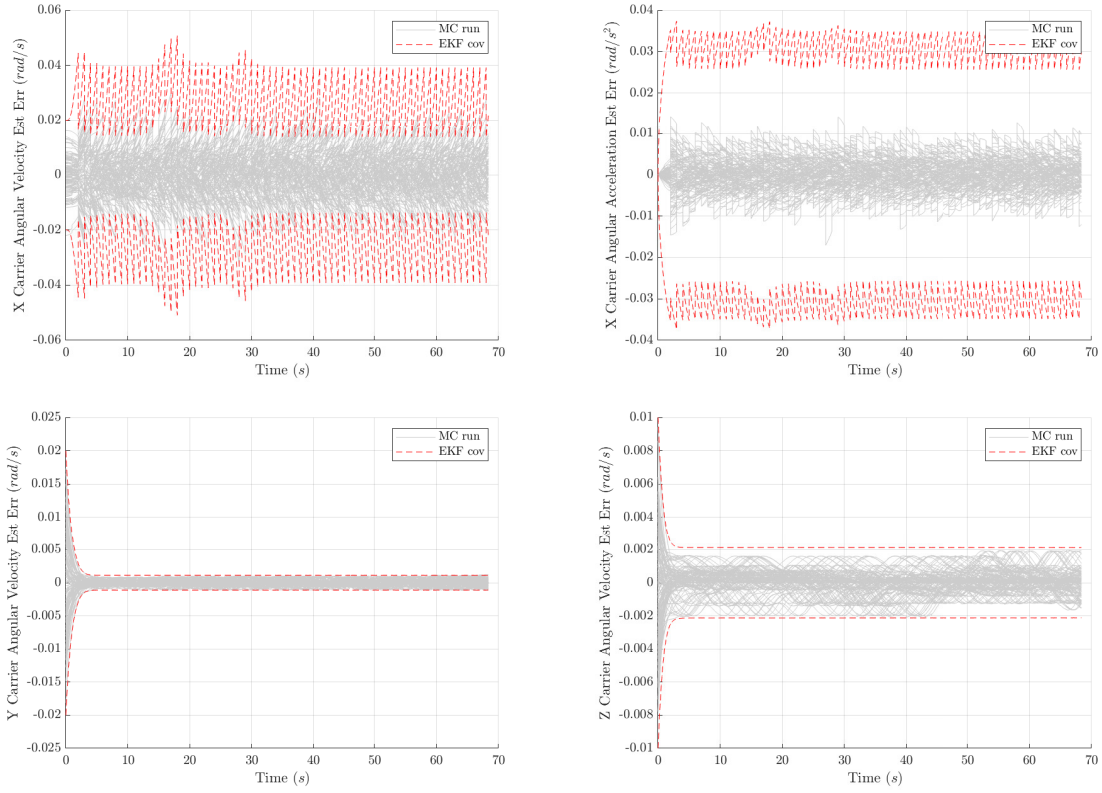

Fig. C.10: Carrier attitude rates estimate

APPENDIX D

Results

D.1 IMU Grade

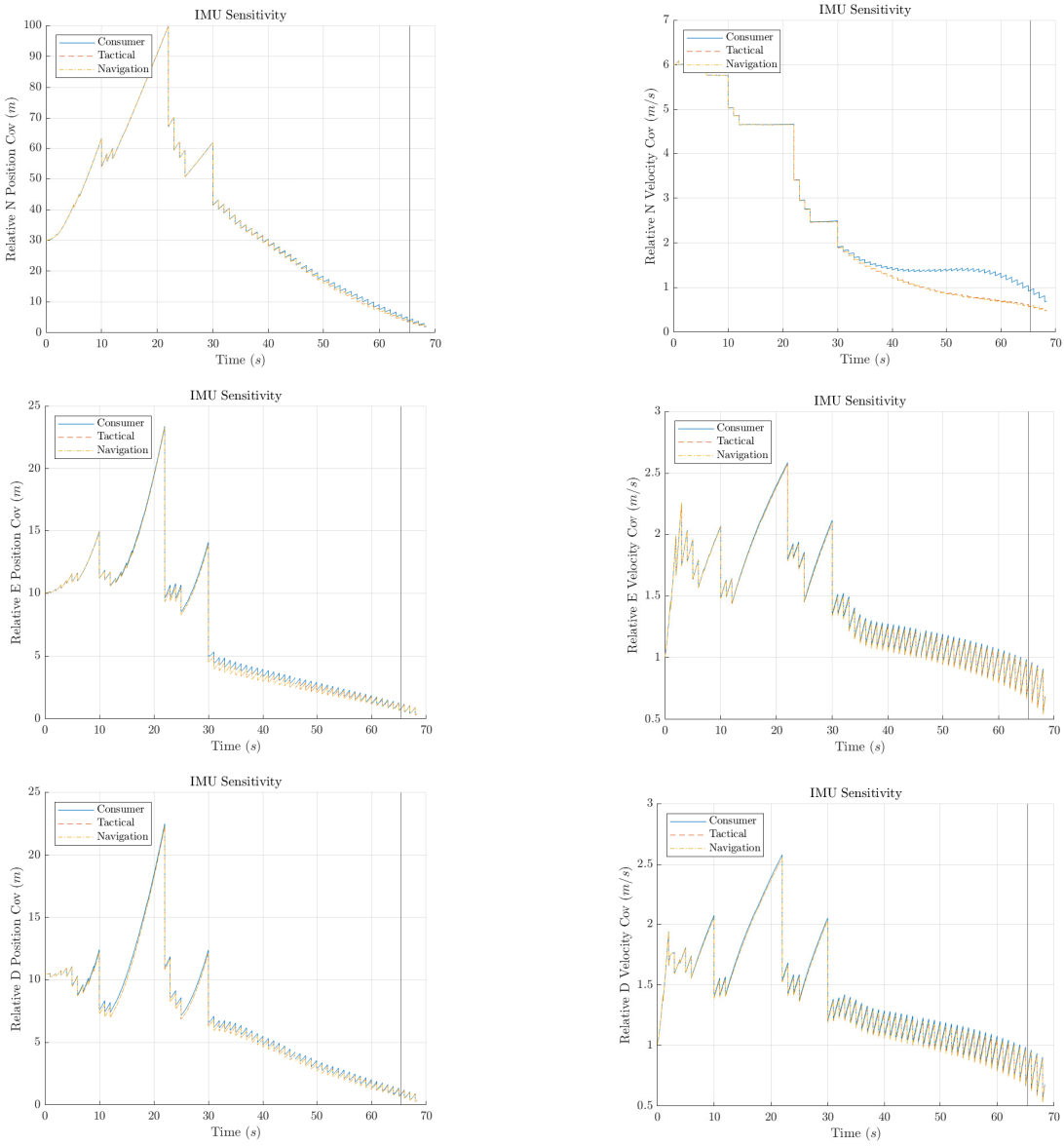

Fig. D.1: IMU study - Relative position and velocity

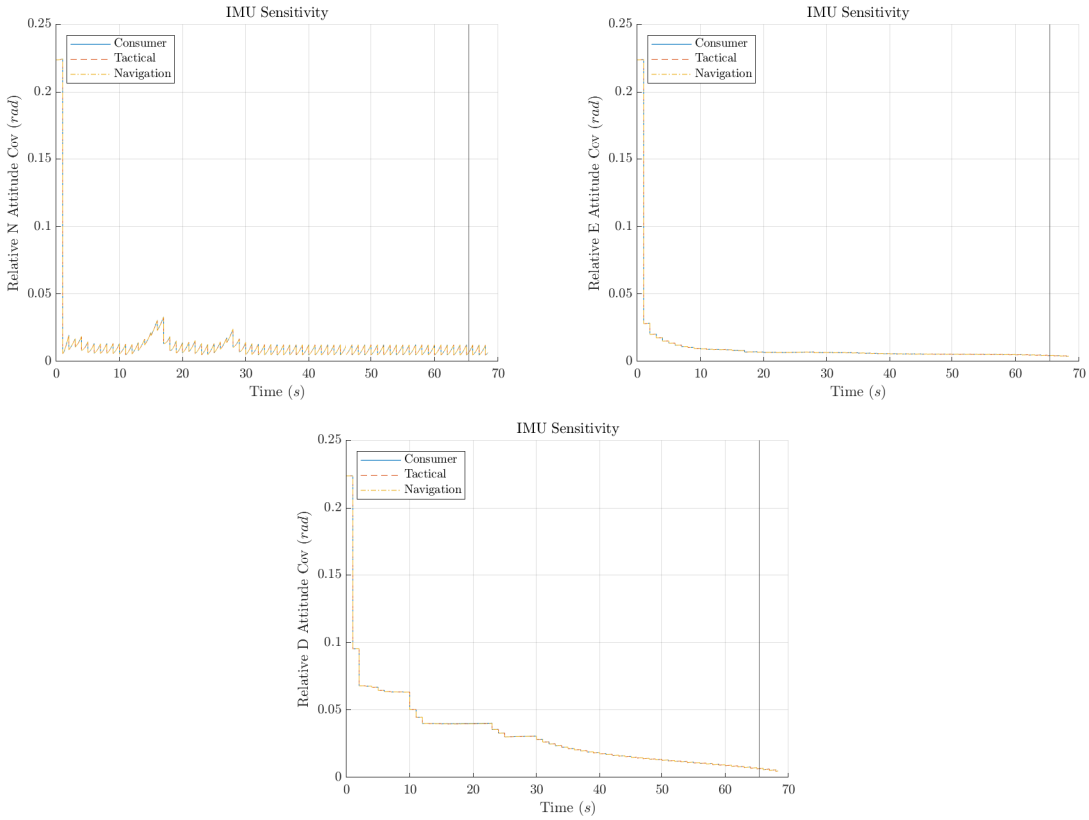

Fig. D.2: IMU study - Relative attitude

D.2 Camera Errors

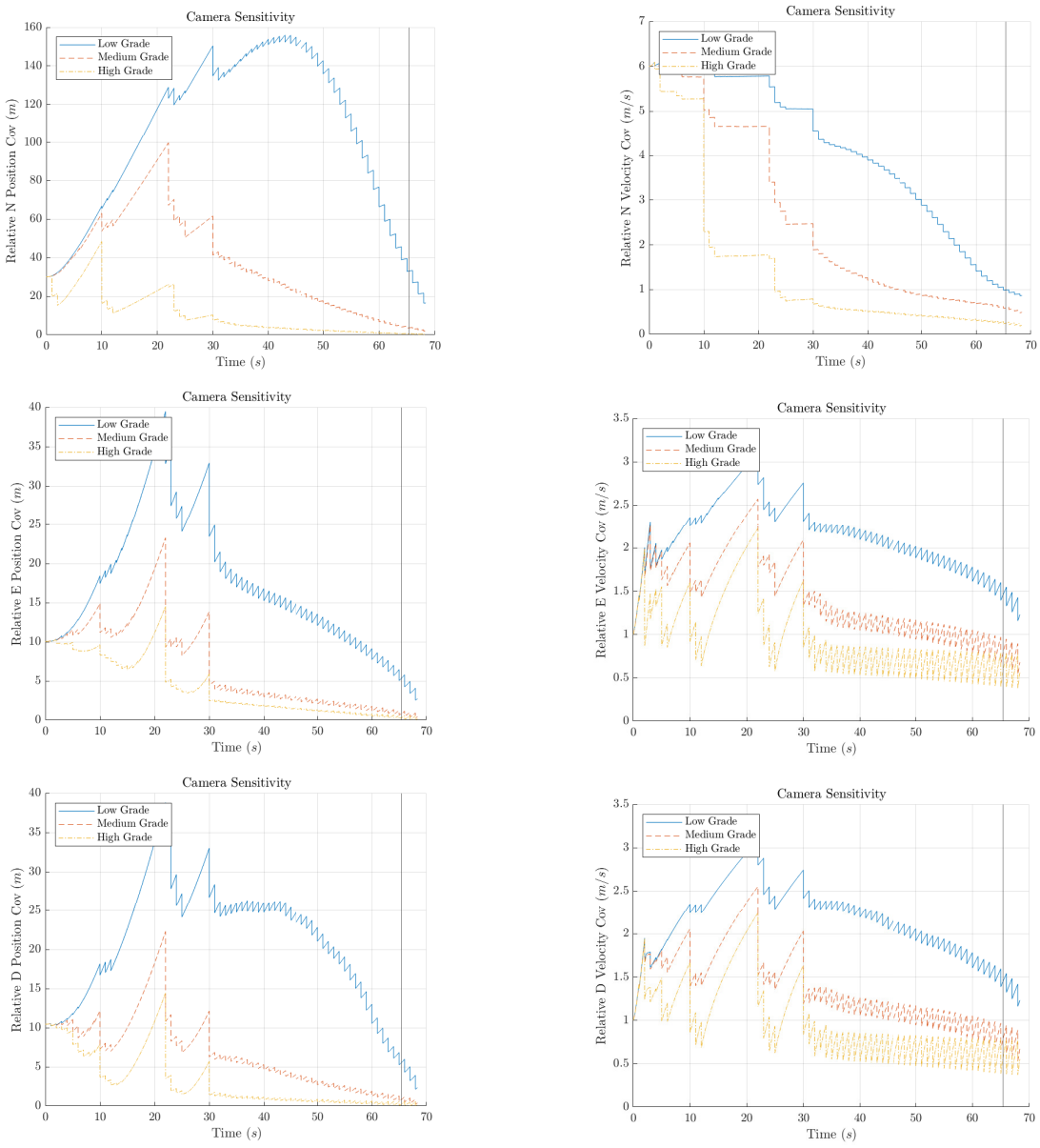

Fig. D.3: Camera study - Relative position and velocity

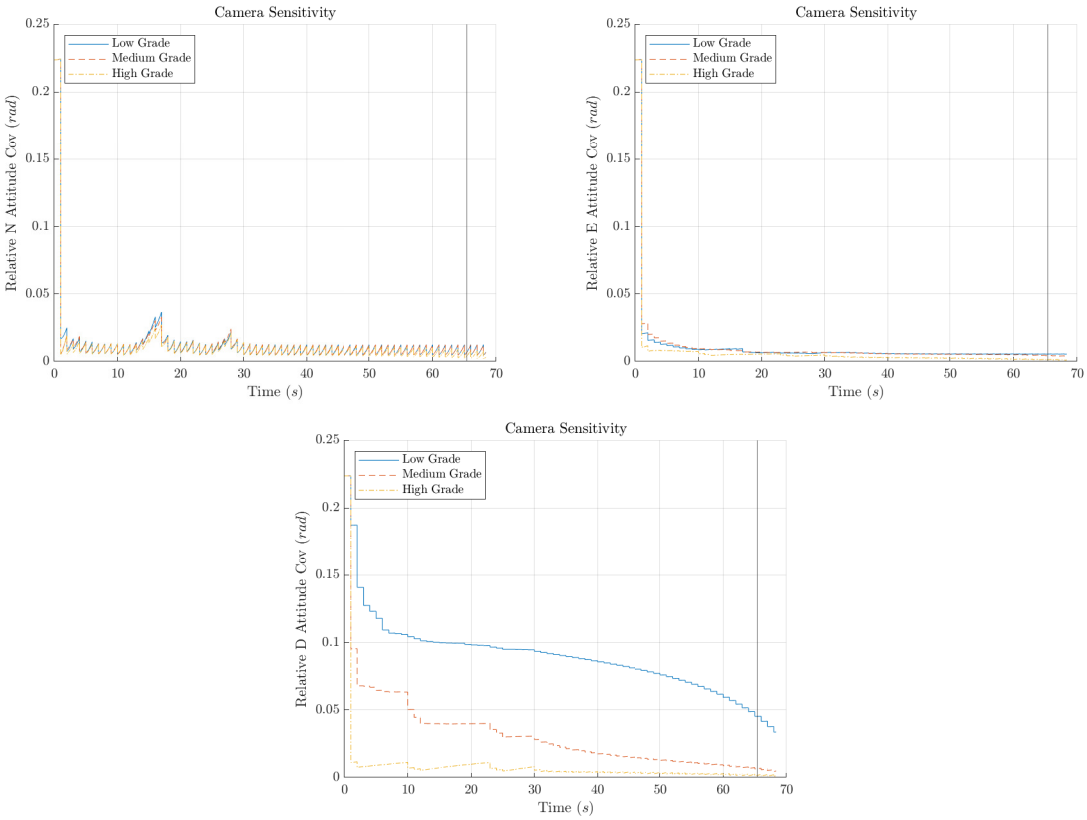

Fig. D.4: Camera study - Relative attitude

D.3 Lidar Errors

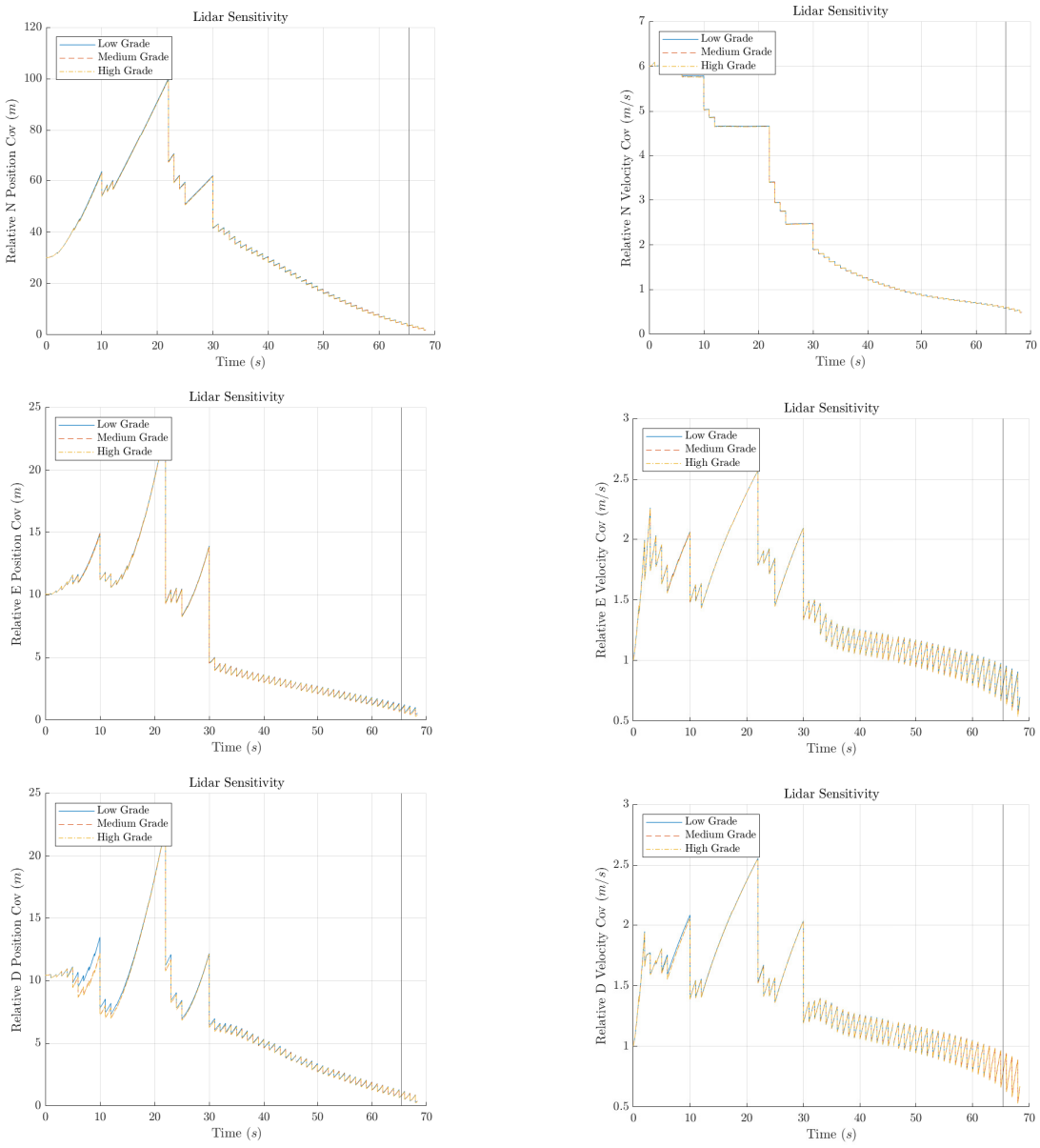

Fig. D.5: Lidar study - Relative position and velocity

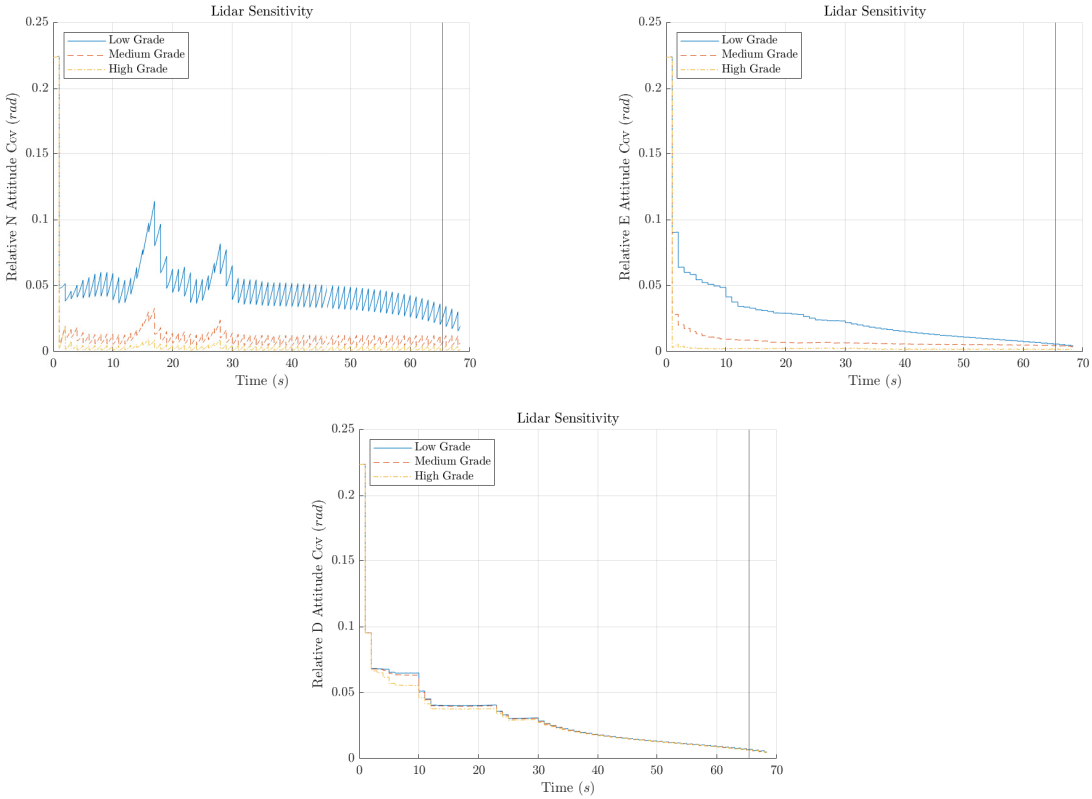

Fig. D.6: Lidar study - Relative attitude

D.4 GPS-denied

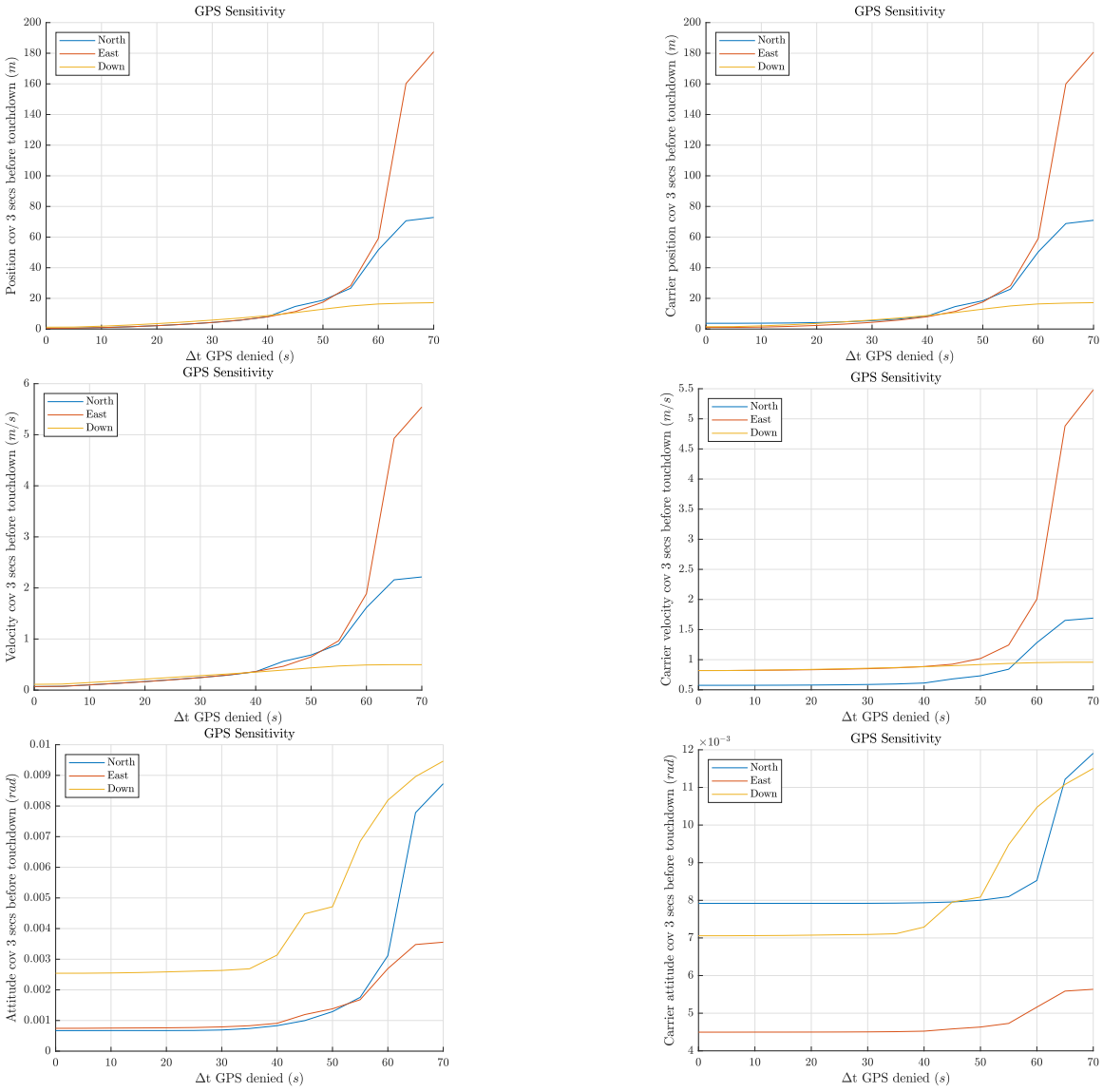

Fig. D.7: GPS study - UAV and carrier position, velocity, and attitude

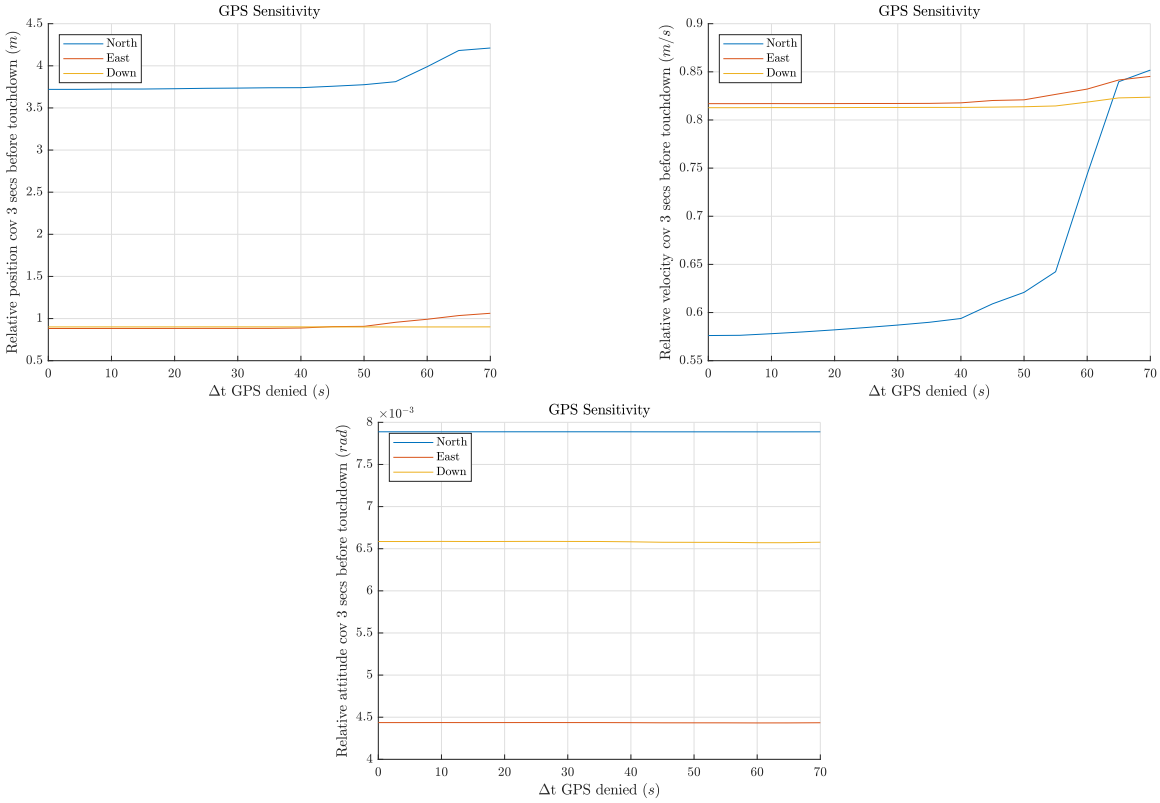

Fig. D.8: GPS study - Relative position, velocity, and attitude

D.5 Fiducial Number

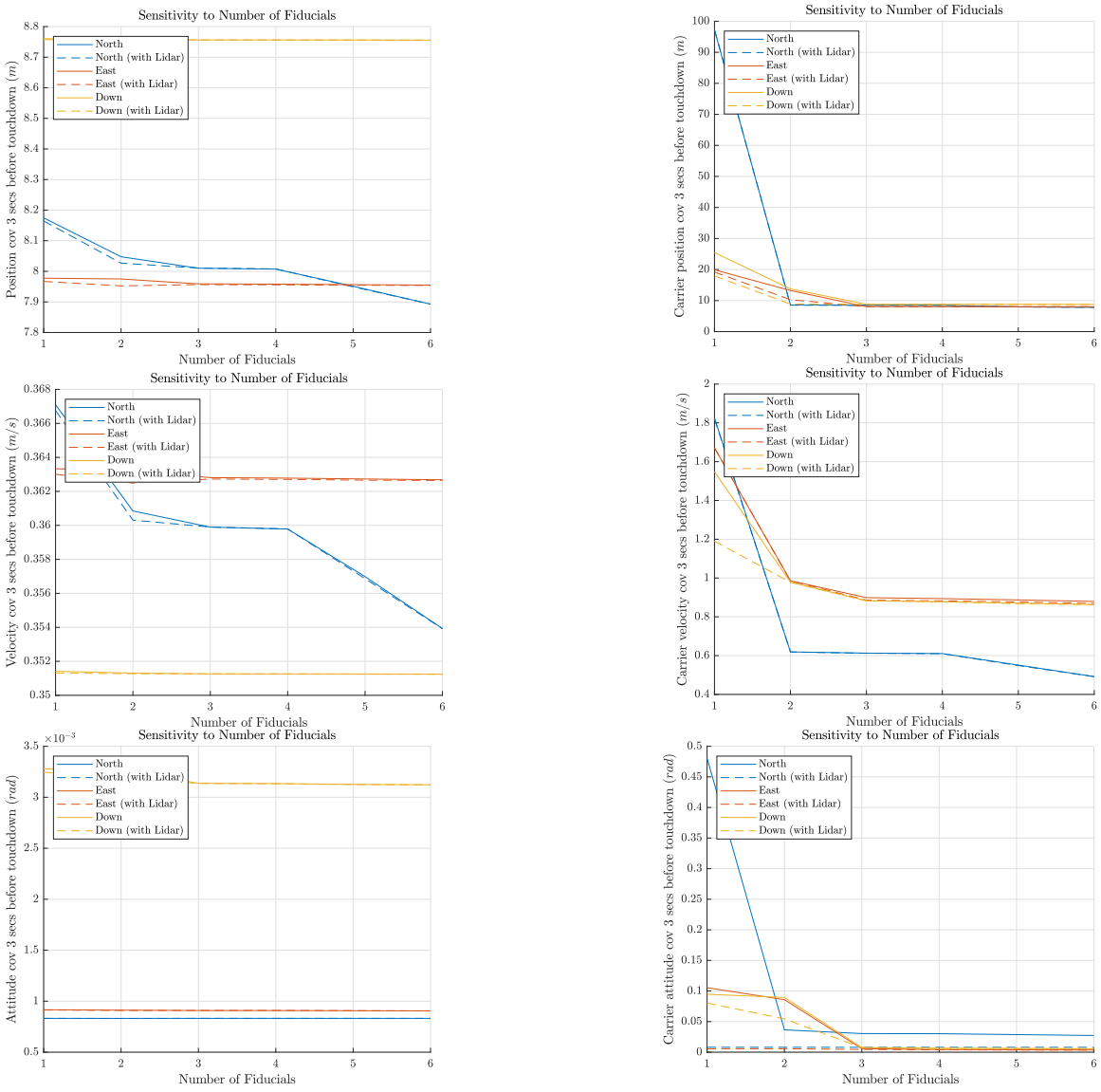

Fig. D.9: Fiducial study - UAV and carrier position, velocity, and attitude

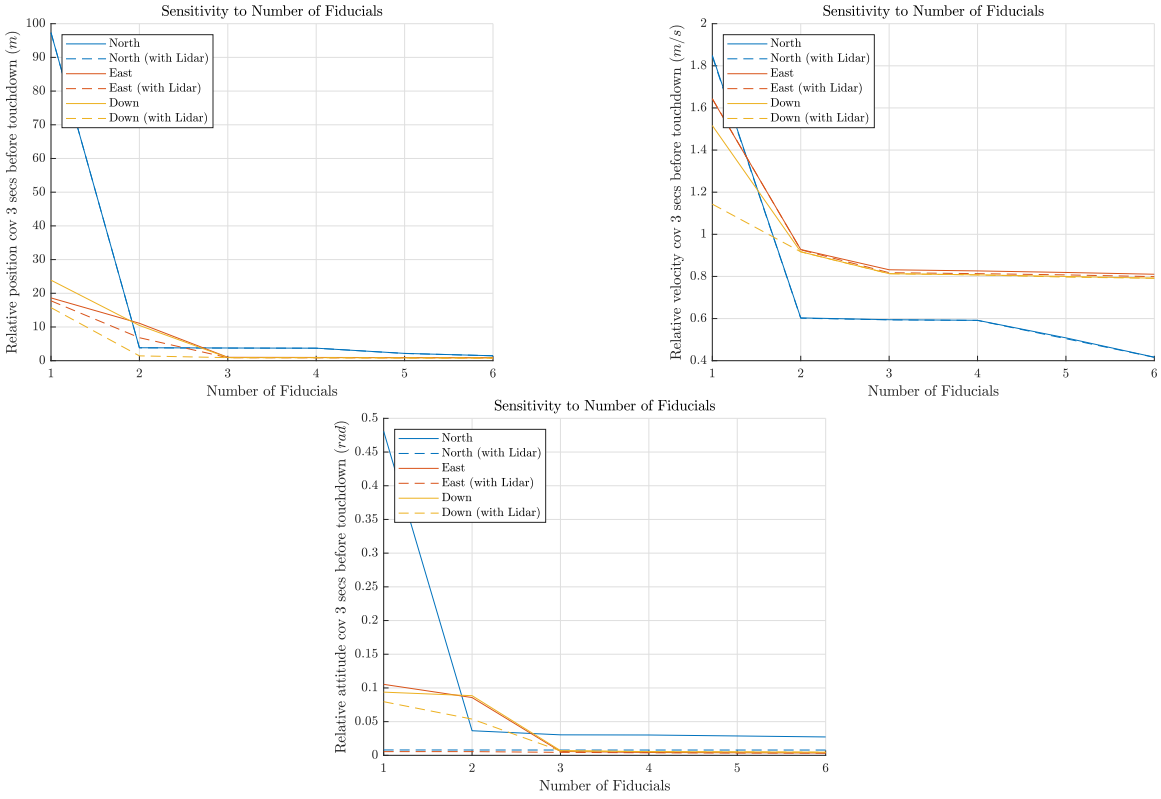

Fig. D.10: Fiducial study - Relative position, velocity, and attitude