SELF-DESCRIBING FIDUCIALS FOR GPS-DENIED NAVIGATION OF UNMANNED AERIAL VEHICLES

by

Amanda J. Strate

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
Jonathan Phillips, Ph.D. Committee Member	D. Richard Cutler, Ph.D. Interim Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2020

ABSTRACT

Self-Describing Fiducials for GPS-Denied Navigation of Unmanned Aerial Vehicles

by

Amanda J. Strate, Master of Science

Utah State University, 2020

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

Accurate estimation of a UAV's location is critical for the safe operation of an autonomous UAV, particularly in GPS-denied environments. Many of the options previously explored for estimation of a UAV's location in GPS-denied environments require large processing resources that may not be available with a processor on a UAV. Many methods are also designed for indoor operation without the ability to be scaled to outdoor operation. This research explores an alternative method of GPS-denied navigation, utilizing line-of-sight measurements to self-describing fiducials, which transmit location information to the observing UAV. In this thesis, an extended Kalman filter based on an inertial navigation system is developed and validated. In simulation, the performance of the estimation algorithm is validated and assessed. The sensitivity of the estimation capability is analyzed by varying the grade of the inertial measurement unit, the density of observed fiducials, the trajectory height above fiducials, and the frequency at which camera measurements were taken. Preparations for demonstrating this system with hardware are presented and discussed. Options for fiducials are analyzed, and ArUco markers are chosen as a viable option for this line of research. Using a motion capture system for obtaining truth data for the states of the UAV is discussed. The noise associated with a motion capture system is measured, and a means of reducing the noise is designed and analyzed. The results from the

simulated scenarios and the hardware demonstration preparation are analyzed, and future work is discussed.

(121 pages)

PUBLIC ABSTRACT

Self-Describing Fiducials for GPS-Denied Navigation of Unmanned Aerial Vehicles

Amanda J. Strate

Accurate estimation of an Unmanned Aerial Vehicle's (UAV's) location is critical for the operation of the UAV when it is controlled completely by its onboard processor. This can be particularly challenging in environments in which GPS is not available (GPS-denied). Many of the options previously explored for estimation of a UAV's location without the use of GPS require more sophisticated processors than can feasibly be mounted on a UAV because of weight, size, and power restrictions. Many options are also aimed at indoor operation without the range capabilities to scale to outdoor operations. This research explores an alternative method of GPS-denied navigation which utilizes line-of-sight measurements to self-describing fiducials to aid in position determination. Each self-describing fiducial is an easily identifiable object fixed at a specific location. Each fiducial relays data containing its specific location to the observing UAV. The UAV can measure its relative position to the fiducial using camera images. This measurement can be combined with measurements from an Inertial Measurement Unit (IMU) to obtain a more accurate estimate of the UAV's location. In this research, a simulation is used to validate and assess the performance of algorithms used to estimate the UAV's position using these measurements. This research analyzes the effectiveness of the estimation algorithm when used with various IMUs and fiducial spacings. The effect of how quickly camera images of fiducials can be captured and processed is also analyzed. Preparations for demonstrating this system with hardware are then presented and discussed, including options for fiducial type and a way to measure the true position of the UAV. The results from the simulated scenarios and the hardware demonstration preparation are analyzed, and future work is discussed.

This work is dedicated to my sweetheart Kaden for his unwavering confidence in me and his never-ending patience. It is also dedicated to my Savior without whose enabling support I would never have made it this far.

ACKNOWLEDGMENTS

I am extremely grateful for the patient guidance and help that my advisor, Dr. Randall Christensen, provided. He encouraged me both in moments of exciting discovery and of epic failure and crashes both of quadcopter and code. I also greatly appreciate the feedback and advice I received from the professors on my committee, Dr. Droge and Dr. Phillips. I am also grateful for my coursework professors who have patiently answered my myriad of questions both in and out of class.

I am also extremely grateful for the support of my family, friends, and classmates. This thesis study would not have been near as rewarding or even possible without their help, support, listening ears, and encouraging words.

Amanda J. Strate

CONTENTS

Page

ABSTRACT . . . iii

PUBLIC ABSTRACT . . . v

ACKNOWLEDGMENTS . . . vii

LIST OF TABLES . . . x

LIST OF FIGURES . . . xi

ACRONYMS . . . xiv

1 INTRODUCTION . . . 1

2 LITERATURE REVIEW AND BACKGROUND . . . 4
- 2.1 Relative State Estimation . . . 4
- 2.2 Absolute State Estimation . . . 5
  - 2.2.1 External Data Processing Methods . . . 6
  - 2.2.2 On-board Data Processing Methods . . . 7
- 2.3 Fiducial-Aided Navigation . . . 8

3 EKF IMPLEMENTATION AND VALIDATION . . . 11
- 3.1 Design Model and Error State Vector Validation . . . 12
- 3.2 Nonlinear State Propagation and Measurement Modeling . . . 17
- 3.3 Linear Error State Modeling . . . 26
- 3.4 Linear Measurement Modeling . . . 30
- 3.5 Covariance Propagation . . . 33
- 3.6 Estimation Capability . . . 49

4 SIMULATION ANALYSIS . . . 59

5 HARDWARE DEMONSTRATION PREPARATION . . . 75
- 5.1 Fiducial Survey and Selection . . . 75
- 5.2 Motion Capture Noise Characterization . . . 79

6 CONCLUSION AND FUTURE WORK . . . 87
- 6.1 Conclusion . . . 87
- 6.2 Future Work . . . 88

REFERENCES . . . 90

APPENDICES 97
A Linearization of State Equations 98
A.1 Position Linearization 98
A.2 Velocity Linearization 99
A.3 UAV Body Frame Linearization 100
A.4 Linearization of Biases and Camera-to-Body Quaternion 103
B Linearization of Measurements 104

LIST OF TABLES

Table Page

3.1 Definition of terms used in the state vectors 13

3.2 MATLAB verification results: error-corrected state subtracted from the truth state 16

3.3 Definition of additional terms for state propagation 18

3.4 $\delta\underline{x}$ in meters calculated using the linear and nonlinear equations and the propagation error difference 29

3.5 Errors injected during MATLAB simulation (Note that g is the acceleration of gravity) 31

3.6 Difference between residual as calculated using the nonlinear equations and the residual as calculated using $H * \delta\underline{x}$ 32

3.7 Difference between X, Y, and Z components of the nonlinear and linear GPS measurement modeling 32

3.8 IMU specification 3-σ values 35

3.9 Initial Uncertainty 3-σ values 35

3.10 Measurement noise 3-σ values 50

4.1 IMU specifications by grade 61

5.1 Standard deviation of position and attitude measurements for the two test configurations of motion capture reflective markers while the UAV was sitting flat on the ground 86

LIST OF FIGURES

Figure Page

1.1 UAV navigating through an urban canyon aided by LOS measurements to SDFs within the camera's FOV . . . 3

3.1 Inertial (I), NED (N), Body (B), and Camera (C) coordinate systems . . . 14

3.2 Frames and relating vectors for state propagation . . . 17

3.3 Simulation UAV trajectory . . . 21

3.4 Position of UAV over time . . . 22

3.5 Position error ($\underline{p}^n - \hat{\underline{p}}^n$) and velocity error ($\underline{v}^n - \hat{\underline{v}}^n$) . . . 22

3.6 UAV attitude error ($\dot{q}_b^n - \hat{\dot{q}}_b^n$) . . . 23

3.7 Accelerometer bias error ($\dot{\underline{b}}_a - \hat{\dot{\underline{b}}}_a$) and gyro bias error ($\dot{\underline{b}}_g - \hat{\dot{\underline{b}}}_g$) . . . 23

3.8 Camera-to-body frame rotation error ($\dot{q}_c^b - \hat{\dot{q}}_c^b$) . . . 24

3.9 Camera measurement error ($\tilde{\underline{z}} - \hat{\tilde{\underline{z}}}$) . . . 25

3.10 GPS measurement error ($\tilde{\underline{z}}_g ps - \hat{\tilde{\underline{z}}}_g ps$) . . . 26

3.11 North, east, and down position estimation error . . . 36

3.12 North, east, and down velocity estimation error . . . 37

3.13 North, east, and down attitude estimation error . . . 38

3.14 X, Y, and Z camera attitude estimation error . . . 39

3.15 X, Y, and Z accelerometer bias estimation error . . . 40

3.16 X, Y, and Z gyro bias estimation error . . . 41

3.17 North, east, and down position estimation error after initial attitude uncertainty adjustment . . . 43

3.18 North, east, and down velocity estimation error after initial attitude uncertainty adjustment . . . 44

3.19 North, east, and down attitude estimation error after initial attitude uncertainty adjustment 45

3.20 X, Y, and Z camera attitude estimation error after initial attitude uncertainty adjustment 46

3.21 X, Y, and Z accelerometer bias estimation error after initial attitude uncertainty adjustment 47

3.22 X, Y, and Z gyro bias estimation error after initial attitude uncertainty adjustment 48

3.23 X and Y camera measurement residuals propagated along with the 3-σ values 51

3.24 X, Y, and Z GPS measurement residuals propagated along with the 3-σ values 52

3.25 North, east, and down position estimation error with Kalman update 53

3.26 North, east, and down velocity estimation error with Kalman update 54

3.27 North, east, and down attitude estimation error with Kalman update 55

3.28 X, Y, and Z camera attitude estimation error with Kalman update 56

3.29 X, Y, and Z accelerometer bias estimation error with Kalman update 57

3.30 X, Y, and Z gyro bias estimation error with Kalman update 58

4.1 Trajectory used to analyze fiducial density and IMU grades 60

4.2 Field of view testing 61

4.3 Estimation error plots for 50- (top), 200- (middle), and 400-meter (bottom) fiducial densities using tactical-grade IMU 63

4.4 Estimation error plots for 200-meter fiducial density using commercial- (top), tactical- (middle), and navigation-grade (bottom) IMUs 64

4.5 Average 3-σ steady state position estimation error for three grades of IMU and varied distance between fiducials for the three axes of the NED frame . 66

4.6 Combined average 3-σ steady state position estimation error for three grades of IMU and varied distance between fiducials 67

4.7 Average 3-σ steady state position estimation error for the three axes of the NED frame as height of the trajectory above the fiducials was varied 69

4.8 Combined average 3-σ steady state position estimation error as height of the trajectory above the fiducials was varied 70

4.9 Estimation error plots for north position estimation with fiducials at 16 m (top) and 18 m (bottom) 71

4.10 Average 3-σ steady state position estimation error for the three axes of the NED frame as frequency of image processing was varied 73

4.11 Combined average 3-σ steady state position estimation error as frequency of image processing was varied 74

5.1 ArUco marker corner sequence reported by ROS ArUco detection software . 77

5.2 Example of ArUco marker image with corners and center marked next to the zoomed-in center of the same image 78

5.3 Additional examples of corner detection and center calculation for ArUco markers 79

5.4 UAV support leg with reflective marker extender beams 80

5.5 Configuration for motion capture reflective markers without extenders . . . 81

5.6 Extended configuration for motion capture reflective markers 81

5.7 Position in the navigation X direction during baseline test for narrow reflector placement (left) and wide reflector placement (right) 82

5.8 Position in the navigation Y direction during baseline test for narrow reflector placement (left) and wide reflector placement (right) 83

5.9 Position in the navigation Z direction during baseline test for narrow reflector placement (left) and wide reflector placement (right) 83

5.10 Attitude rotation about the navigation X axis during baseline test for narrow reflector placement (left) and wide reflector placement (right) 84

5.11 Attitude rotation about the navigation Y axis during baseline test for narrow reflector placement (left) and wide reflector placement (right) 84

5.12 Attitude rotation about the navigation Z axis during baseline test for narrow reflector placement (left) and wide reflector placement (right) 85

ACRONYMS

BVLOS	beyond visual line-of-sight
EKF	extended Kalman filter
FAA	federal aviation administration
FOV	field of view
GPS	global positioning system
IMU	inertial measurement unit
LOS	line-of-sight
RFID	radio-frequency identifiers
ROS	robot operating system
SDF	self-describing fiducial
SLAM	simultaneous location and mapping
UAV	unmanned aerial vehicle
VLOS	visual line-of-sight
VO	visual odometry
WLAN	wireless local area network

CHAPTER 1

INTRODUCTION

Interest in commercial Unmanned Aerial Vehicle (UAV) use has grown over the last several years, including for package delivery. However, many obstacles must be overcome before drones (UAVs) can be feasibly and safely operated autonomously for this purpose [1]. Drones need to comply with rules and regulations set forth by governing agencies such as the United States' Federal Aviation Administration (FAA). These agencies require UAVs to consistently operate within strict specifications. If a drone were to vary from these specifications, it could have dire consequences such as interfering with the flight of a passenger aircraft or other public services. One such specification that limits the broad adoption of UAV package delivery is the requirement to operate the UAV within Visual Line-of-Sight (VLOS) of the pilot operating the drone. Although Beyond Visual Line-of-Sight (BVLOS) operation is becoming a possibility, the process to acquire a waiver is difficult and requires guarantees of safe operation of the UAV in a variety of challenging circumstances [2].

One particularly challenging situation encountered by UAVs is operation in GPS-denied or GPS-challenged environments. Degradation or loss of GPS can occur in a variety of situations, both indoor and outdoor. Operation in deep urban or natural canyons is often associated with reflected GPS signals, causing multipath corruption of the GPS signals reaching the receiver. Canyons can also reduce the number of visible satellites and increase the geometric dilution of precision of position measurements obtained from visible satellites [3], [4]. GPS signals can also be intentionally jammed or spoofed [5]. Without a suitable source of position information, prolonged GPS signal outages can result in a large divergence from the intended flight path, and the results can be catastrophic. Even when GPS data is available, GPS measurements can be off by several meters [6], [7]. Depending on the application, absolute position updates may need to be more accurate than is available

through GPS.

This research consists of designing and analyzing a low-cost, ubiquitous replacement for GPS, consisting of carefully-placed fiducials along the path of the UAV and an associated image processing system, inertial navigation system, and Extended Kalman Filter (EKF). The "self-describing fiducials" (SDFs) are designed to transmit their locations to the observing UAV, thus providing an independent source of positioning data as proposed in Davidson et al. [8].

The three objectives of this research were:

1. Develop an EKF to combine line-of-sight (LOS) measurements to self-describing fiducials with inertial measurement unit (IMU) measurements. Validate the EKF in a Monte Carlo simulation.

2. Study the sensitivity of estimation errors to varying IMU grades, fiducial spacing, UAV trajectory height above fiducials, and image processing frequency.

3. Make preparations for hardware implementation of the EKF.

This research provides insight into the effectiveness and required configuration for using SDFs as a replacement for GPS signals when GPS signals are not available or are degraded. A pictorial depiction of the proposed SDF system operating in an urban canyon is shown in Figure 1.1 [9], [10], [11]. In this system, the UAV processes fiducials that are within the camera's field of view (FOV) as it proceeds between tall buildings that distort or completely block available GPS signals.

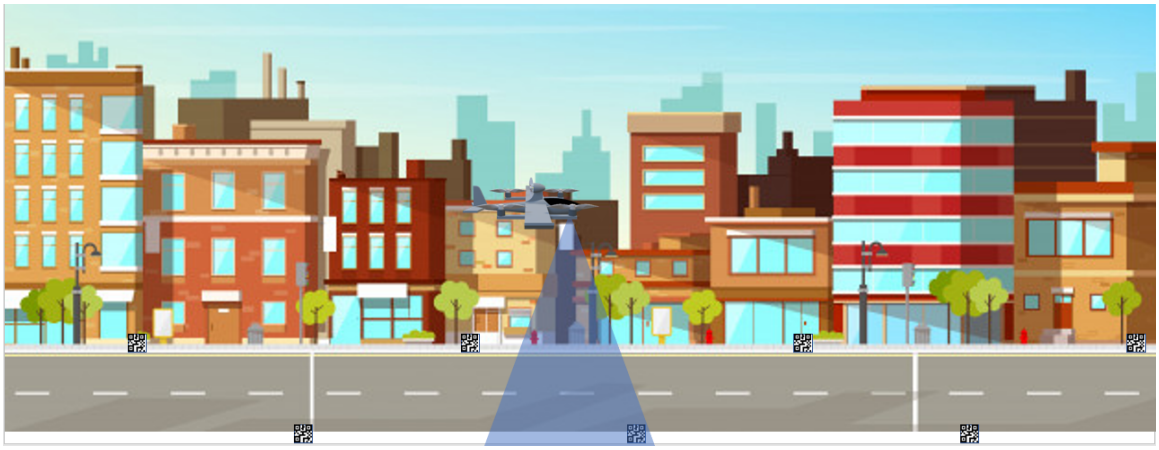

Fig. 1.1: UAV navigating through an urban canyon aided by LOS measurements to SDFs within the camera's FOV

The rest of this thesis is organized as follows. Chapter 2 discusses previous research performed in the general field of GPS-denied navigation as well as the more specific discipline of using fiducials for navigation. Chapter 3 shows the development and validation of the EKF used in this study. Chapter 4 presents and discusses the results of simulating navigation of a UAV aided by fiducial measurements as various parameters are adjusted such as the quality of IMU employed for accelerometer and gyroscope measurements. Chapter 5 discusses a possible fiducial type and a method for obtaining truth data for hardware implementation of this research. In Chapter 6, results of this research and areas for future work are discussed.

CHAPTER 2

LITERATURE REVIEW AND BACKGROUND

The many possible applications of autonomous drone flight have drawn considerable attention to the problem of GPS-denied navigation. Many strategies have been applied to the challenge, and they can be separated into two categories: relative and absolute. Relative methods determine the current position of a UAV by measuring its change in position relative to a previous estimate. Because the next position is determined solely based on previous states and measurements of change from that position, the position uncertainty grows over time without absolute location updates to bind the errors. Absolute methods employ some means of correcting the error in position estimation that grows over time. This requires an external source of known position data which provides reference points along the UAV's trajectory. These reference points can be used by the UAV to determine its location.

2.1 Relative State Estimation

One relative approach that has been heavily researched is simultaneous location and mapping (SLAM) [12], [13], [14], [15], [16], [17], [18]. SLAM methods reduce the UAV state estimation errors by using camera images to track salient features between images. While this method can approach absolute state estimation when a library of known landmarks and associated positions is used [19], any errors in mapping the surroundings can cause accumulated state estimation error over time. The computational requirements are high, however, due to the large amount of image processing associated with each frame as well as the high number of filter states required to map the environment. Because of this, processing must be done in a closed-loop system using a computer not located on the UAV to process the data and maintain the library of features. Wireless communication may not be possible in outdoor GPS-denied environments, preventing the use of off-board computers

for processing images, thus deeming the SLAM approach unfeasible.

Another relative approach is that of visual odometry (VO) [20], [21]. VO employs an algorithm that exploits changes in camera images from frame to frame to reduce the estimation error of an inertial navigation system but does not attempt to map the surroundings, thus avoiding the high computational requirements [22]. VO is particularly effective when combined with Inertial Measurement Unit (IMU) data through an Extended Kalman Filter (EKF) [23], [20].

Multiple studies have shown that using two cameras rather than a single monocular camera can improve the effectiveness of both the SLAM and VO approaches [22]. However, these approaches still cannot fully provide absolute state estimate updates. These approaches are also highly dependent on lighting conditions [24].

Multiple possible sensing modalities could be used to estimate the change in position of a UAV between time frames. No matter the sensor(s) used, if a method for re-establishing a known position or state is not employed, the uncertainty of the UAV's state will continue to grow over time, critically limiting the time for which the estimated state can be considered to be reliable.

2.2 Absolute State Estimation

Any state estimation method that has the capability to provide absolute state estimation updates requires external infrastructure of some sort. These methods are broken into two categories in this review: methods that require external data processing and methods which allow for all data processing to be done on-board. Another way of phrasing this is to say that the external infrastructure is either active (external data processing) or passive (on-board data processing). Passive external infrastructure simply has to be present for the system to work. The UAV senses the external infrastructure and generates its own measurements to absolute locations. Active infrastructure requires the capability to process data about the UAV's position and then provide information to the UAV through some form of active communication.

2.2.1 External Data Processing Methods

Perhaps the most common method used for absolute state estimation is using GPS signals. As mentioned in the introduction to this thesis, GPS signals often only provide accuracy of a few meters [25], [7]. A lot of effort has been put forth to increase the accuracy of GPS signals by either devising new means of processing the signals such as in Yun et al. [26] or by combining GPS signals with inertial sensors [27], [28], [29], [30]. However, as mentioned, GPS signals are not always available or reliable.

One common active method used indoors to track the state of a UAV is using a motion capture system [31], [32], [33], [34], [35]. Identifying markers are placed on the drone such as reflective orbs, and several cameras are placed around the UAV's flight area which monitor the location of the identifying markers and therefore the UAV. The gathered location data is then sent to the flight controller. This approach can attain estimation error in the micrometer range [36]. While this method is extremely accurate, it requires significant additional infrastructure and time investment. The cameras have to be arranged in a manner that provides thorough coverage from all angles without objects being between the cameras and the drone. Each camera also needs to have a connection to a common network so that the data from each camera can be fused and then sent to the flight controller. The cameras also need to be calibrated regularly both for the space they encompass as well as for the markers on the UAV. The infrastructure required can also be quite expensive [37].

Another active indoor localization approach is to use data transmission times between multiple devices at know locations to triangulate the position of the UAV [38]. For example, a wireless local area network (WLAN) has been shown to be an effective active method for locating objects without the aid of GPS [39], [40], [41]. Measurements of the wireless network's signal strength are taken at various points within the area of interest. These measurements can then be compared against the signal strength detected by a navigating object to determine a likely position at which the object is located. During the calibration process, if similar signal strengths are detected at multiple locations, further analysis must be employed to determine location data. This method requires an existing network and often

has comparatively large position estimation errors. Bluetooth signals can be used similarly, but Bluetooth signal range is much less than that obtainable with WLAN [42], [43]. While the network does not directly provide position estimates to the UAV, data must still be actively transmitted from the infrastructure (the WLAN system) to the UAV in order for the UAV to estimate its position.

2.2.2 On-board Data Processing Methods

Localization methods that allow for all data to be processed on-board the UAV come with several benefits. One is that the position estimation is not dependent on active data communication between the UAV and an off-board computer. The UAV senses specific locations provided by the external infrastructure and generates state estimations from this without any data needing to be processed externally.

An example of this approach is using radio-frequency identification (RFID) tags to track the location of an object [44], [45], [46], [43]. When passive RFID tags are placed in specific locations in the UAV's environment, the UAV can sense these tags and therefore estimate its position without the external infrastructure needing to actively transmit data to the UAV. A drawback to this method is that it does not provide either the accuracy or the range that would be required to localize a UAV, particularly in outdoor environments. Each of these discussed absolute approaches require that power be supplied to the infrastructure, resulting in a large energy footprint, which is one of the main concerns mentioned in Stolaroff [1].

The method which is the focus of this research, using LOS measurements to self-describing fiducials, also uses a passive external infrastructure. This external infrastructure consists of fiducials that are strategically placed in the UAV's environment. The UAV need only sense its relative location to these fiducials without the need for any external infrastructure to sense the UAV or actively send data to the UAV. If the fiducials used do not require power to represent data, this method does not require power for the external infrastructure. This method is discussed in more detail in Section 2.3.

2.3 Fiducial-Aided Navigation

Recently, a series of carefully-placed fiducials was proposed as a solution to robust UAV state estimation in GPS-degraded or -denied environments [8]. The "self-describing fiducials" are designed to transmit their location relative to a local or global coordinate system. The fiducials could contain all of the data about their position within their patterns, or they could transmit an identifier to the UAV. A table of identifiers and the associated locations would be stored on the UAV so that it could look up the fiducial ID and therefore have its location data. The UAV observes the fiducials with an on-board camera, decodes the position information, and processes the observation as LOS measurements to the fiducials with known locations. The result is an aided inertial navigation system that provides absolute state estimation updates.

In addition to constraining the estimation error, navigation using SDFs provides several other benefits that the other error-bound methods do not provide. Many types of fiducials do not consume any power, which would leave a much smaller energy footprint. Many fiducials such as AprilTags can also provide localization with six degrees-of-freedom [47]. Multipath errors do not occur with the SDF system because the fiducials must be within LOS of the UAV's camera to be read, and the image cannot be read from any means other than a direct line. The range at which these measurements can be taken is limited only by the resolution of the camera, the type and size of fiducial chosen, and the view angle of the camera.

Another benefit associated with SDF-aided navigation is that processing the fiducials is algorithmically simple enough that it can be done with an on-board processor. Also, because the SDFs are artificial fiducials that are associated with limited data, the memory requirements for this method are smaller than would be required for methods such as SLAM. Reduced processing and memory requirements required for SDF-aided navigation allow for all of the processing to be done on-board the UAV without the need to off-load the processing to a ground station.

Previous implementations of a similar system have indicated that it is a feasible solution to GPS-denied navigation. In Gerratt [48], Nguyen et al. [49], and Yu et al. [50], fiducials have been shown to effectively aid in the landing of UAVs. The effectiveness of SDFs in UAV navigation is displayed even further in Kahani et al. [51], where camera measurements of AprilTag (a type of fiducial) positions relative to the UAV were fused with IMU measurements through an EKF. The effectiveness of the approach was analyzed as the distance to the AprilTags as well as the view angle of the fiducials was varied. The effectiveness of using only one AprilTag compared to using three was also compared. It was observed that a root mean square error of 0.20 m was attainable when all three AprilTags were used.

After the work which exhibited the concept of navigation using AprilTags in Nahangi et al. [52] and Kahani et al. [51], Kahani et al. summarized key challenges that still need to be attended to [53]. Some challenges discussed include those of partially- or fully-occluded fiducials, calibration of fiducial placement, type of fiducial, and tag distribution.

Several other researchers have done research that helps to tackle some of these challenges. Some have tackled the problem of partially occluded or blurred images of the fiducials [54], [55]. Many others have explored the effectiveness of various types of fiducials for the use of localization including AprilTags [47], [56], [57], ARTag [58], [59], CALTag [60], QR Codes [61], and even LEDs. Using LEDs as fiducials is beneficial because they are less dependent on lighting conditions [62], [63]. The detection of fiducials in natural, less controlled environments has been researched, including various lighting environments [64] and underwater environments [65].

A key area in which current research is still lacking is that of the precise system requirements to achieve specific bounds of error on a UAV's state estimation using SDFs for navigation. The previously-discussed research does not address how closely the fiducials must be spaced and the quality of IMU that must be paired with the fiducial measurements to achieve effective results. A study in Cavanini et al. [61] used SDFs on the ceiling for the localization of a smart wheelchair and stated that fiducials needed to be placed more

than once per square meter. However, there was not much analysis beyond this simple recommendation on the topic of fiducial density, fiducial measurements were not paired with IMU measurements, and the study was limited to a device that could only navigate in a two-dimensional plane. Previous studies have also focused on indoor use, while this research seeks to broaden the use to include outdoor environments.

This thesis research aimed to fill in part of this lacking area and provide more insight into the fiducial density, IMU quality, and other factors required to obtain specific bounds on the state estimation error of a UAV employing both IMU and fiducial measurements through an EKF. While previous fiducial-aided navigation studies have focused on specific types of fiducials, this research seeks to provide a general analysis that could be applied to many types of fiducials by focusing on single-point fiducials. Each of the fiducial types discussed in previous studies can be reduced to a single point in the center of the fiducial. While each fiducial type would introduce different variables such as distance at which the fiducial can accurately be read and the amount of data that can be transmitted by the fiducial, the principles in this study can be used as guidelines for possible setup requirements for each.

CHAPTER 3

EKF IMPLEMENTATION AND VALIDATION

In this chapter, the EKF developed for the purpose of combining IMU measurements with GPS and LOS measurements to available fiducials is discussed. The nonlinear equations that determine the state vector are developed, and these equations are linearized for use in the Kalman filter. The bias and noise associated with the accelerometer and gyroscope in the IMU are accounted for in the equations. The EKF is validated systematically through progressive verification steps.

A Kalman filter is a method of predicting state variables such as the position and velocity of a body in motion. It does this by using a history of past states along with measurements of the states over time. It uses the statistical variance of each of the measurements to determine how likely a state is given a specific measurement. This information is then used along with previously-estimated states to update its estimation of the current states of the body in motion. A Kalman filter can only update covariance of states using linearly-related variables. An Extended Kalman Filter (EKF) is a Kalman filter that can work with nonlinear equations. The nonlinear equations are linearized and then used in a Kalman filter.

Section 3.3 presents the design model of the UAV system along with nonlinear equations used to relate truth state estimates to estimated state estimates as well as the error state difference between the two. This error state vector is validated. The coordinate frames used in this study are also presented and discussed. Section 3.2 uses the nonlinear state equations along with pre-calculated IMU measurements throughout a test trajectory to propagate the states of the UAV. This propagation is used in later steps for both the navigation state (the estimated state of the UAV using sensor measurements) and the truth state. This section verifies that the truth and navigation states are propagated using the nonlinear equations with no difference between the two before error is added into the navigation state in later

sections.

To obtain linear equations on which a Kalman filter can operate, the nonlinear equations are linearized in Section 3.3. The navigation states are then propagated using these linear equations, and the propagation is compared to the propagation using the nonlinear equations to verify that the difference between the two methods of propagation is sufficiently small. Section 3.4 discusses the linearization of the GPS measurements as well as the camera measurements. Small estimation errors are injected into both the nonlinear models as well as the linear models, and the measurements using both models are compared. The difference is verified to be sufficiently small.

In Section 3.5, errors are injected into the IMU measurements so that the estimated state propagation no longer exactly matches the truth state propagation. Monte Carlo simulations are run to verify that as the state propagation is done many times, the resulting errors are zero-mean and stay mostly within 3-σ bounds. In Section 3.6, measurement updates from GPS and fiducial measurements are added to aid in the state estimation. This is when the Kalman filter begins to really be useful. GPS measurements are used for the first portion of the state propagation, and fiducial measurements are used during the last portion after GPS measurements are lost.

3.1 Design Model and Error State Vector Validation

The states used for the model and simulation of the UAV's flight dynamics are discussed here. This section shows the relationships between the truth (reality), navigation (estimated states), and error states (the difference between the truth and navigation states) resulting from the injected error. In later sections, error will be injected into both the continuous (IMU) and discrete (GPS and camera) measurements. Before this is done, the equations used to relate the truth, navigation, and error states need to be validated. This is done in this section. Terms used in this section are defined in Table 3.1. Note that $\underline{\ }$ denotes a vector, such as with $\underline{x}$. Also, note that $\hat{\ }$ denotes the estimated value of a variable.

The truth state vector is constructed as follows:

$$\underline{x} = [\quad \underline{p}^n \quad \underline{v}^n \quad q_b^n \quad \underline{b}_a \quad \underline{b}_g \quad q_c^b \quad]^T \tag{3.1}$$

Term	Definition	Units
$\underline{p}^n$	Position of the UAV with respect to the inertial frame as expressed in the North East Down (NED) frame, also known as the navigation frame	m
$\underline{v}^n$	Velocity of the UAV with respect to the inertial frame as expressed in the NED frame	m/s
q_b^n	Quaternion from the body frame to the NED frame	unitless
$\delta\theta_b^n$	Angular difference between the estimated $\hat{q}_b^n$ and the true q_b^n	$radians$
$\underline{b}_a$	Bias of the accelerometer	m/s^2
$\underline{b}_g$	Bias of the gyroscope	$radians/s$
q_c^b	Quaternion from the camera frame to the body frame	unitless
$\delta\theta_c^b$	Angular difference between the estimated $\hat{q}_c^b$ and the true q_c^b	$radians$

Table 3.1: Definition of terms used in the state vectors

The navigation state vector (denoted by $\hat{}$) is constructed as follows:

$$\hat{\underline{x}} = [\quad \hat{\underline{p}}^n \quad \hat{\underline{v}}^n \quad \hat{q}_b^n \quad \hat{\underline{b}}_a \quad \hat{\underline{b}}_g \quad \hat{q}_c^b \quad]^T \tag{3.2}$$

The error state vector is constructed as follows:

$$\delta\underline{x} = [\quad \delta\underline{p}^n \quad \delta\underline{v}^n \quad \delta\underline{\theta}_b^n \quad \delta\underline{b}_a \quad \delta\underline{b}_g \quad \delta\underline{\theta}_c^b \quad]^T \tag{3.3}$$

There are four coordinate systems (frames) for UAVs that apply to fiducial-guided UAV navigation: the inertial, North East Down (NED or N), body, and camera frames. These are shown in Figure 3.1 [9]. The inertial frame is in reference to a fixed point on the ground. The duration of a UAV flight is considered to be brief enough that the effects of the variation of the magnitude of the acceleration due to gravity as well as the effects of the rotating earth are considered to be insignificant. Similar navigation frameworks can be found in Grewal et al. [66], Farrell [67], and Savage [68]. The framework most similar to the framework used in this research is the "tangent frame" in Farrell [67]. The x, y, and z axes of the inertial frame are fixed. The NED frame is parallel with the inertial frame (axes are also fixed), but its origin is at the center of mass of the UAV. The body frame also has its origin at the center of mass of the UAV, but its axes rotate with the UAV. The x-axis points out the front of the nose, the y-axis points out toward the right wing, and the z-axis

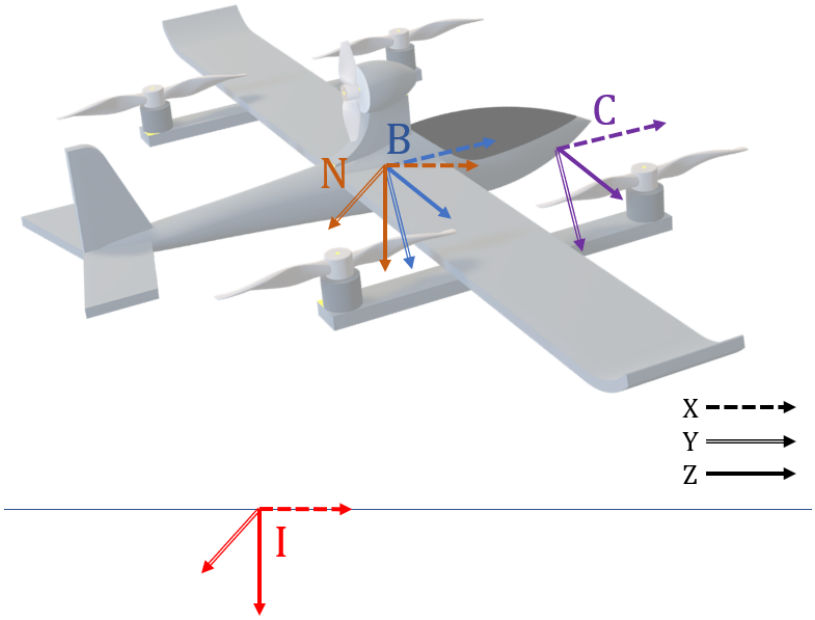

Fig. 3.1: Inertial (I), NED (N), Body (B), and Camera (C) coordinate systems

points out the bottom of the fuselage of the UAV. The camera frame is at a fixed rotation in relation to the body frame and offset by a lever arm.

The relationships between the truth, navigation, and error state vectors are shown in equations 3.4, 3.5, and 3.6. This mapping shows the relationships required to go from truth

state, to a navigation state, to the error state, and back to the truth state.

$$\underline{x} = \begin{bmatrix} \underline{p}^n \\ \underline{v}^n \\ q_b^n \\ \underline{b}_a \\ \underline{b}_g \\ q_c^b \end{bmatrix} = \begin{bmatrix} \hat{\underline{p}}^n + \delta\underline{p}^n \\ \hat{\underline{v}}^n + \delta\underline{v}^n \\ \begin{bmatrix} 1 \\ -\frac{1}{2}\delta\underline{\theta}_b^n \end{bmatrix} \otimes \hat{q}_b^n \\ \hat{\underline{b}}_a + \delta\underline{b}_a \\ \hat{\underline{b}}_g + \delta\underline{b}_g \\ \begin{bmatrix} 1 \\ -\frac{1}{2}\delta\underline{\theta}_c^b \end{bmatrix} \otimes \hat{q}_c^b \end{bmatrix} \tag{3.4}$$

$$\hat{\underline{x}} = \begin{bmatrix} \hat{\underline{p}}^n \\ \hat{\underline{v}}^n \\ \hat{q}_b^n \\ \hat{\underline{b}}_a \\ \hat{\underline{b}}_g \\ \hat{q}_c^b \end{bmatrix} = \begin{bmatrix} \underline{p}^n - \delta\underline{p}^n \\ \underline{v}^n - \delta\underline{v}^n \\ \begin{bmatrix} 1 \\ -\frac{1}{2}\delta\underline{\theta}_b^n \end{bmatrix}^* \otimes q_b^n \\ \underline{b}_a - \delta\underline{b}_a \\ \underline{b}_g - \delta\underline{b}_g \\ \begin{bmatrix} 1 \\ -\frac{1}{2}\delta\underline{\theta}_c^b \end{bmatrix}^* \otimes q_c^b \end{bmatrix} \tag{3.5}$$

State	$\underline{p}^n$	$\underline{v}^n$	q_b^n	$\underline{b}_a$	$\underline{b}_g$	q_c^b
Difference X	0	0	0	0	0	1.36e-16

Table 3.2: MATLAB verification results: error-corrected state subtracted from the truth state

$$\delta\underline{x} = \begin{bmatrix} \delta\underline{p}^n \\ \delta\underline{v}^n \\ \delta\underline{\theta}_b^n \\ \delta\underline{b}_a \\ \delta\underline{b}_g \\ \delta\underline{\theta}_c^b \end{bmatrix} = \begin{bmatrix} \underline{p}^n - \hat{\underline{p}}^n \\ \underline{v}^n - \hat{\underline{v}}^n \\ -2\delta\hat{q}_b^n(2:4) \\ \underline{b}_a - \hat{\underline{b}}_a \\ \underline{b}_g - \hat{\underline{b}}_g \\ -2\delta\hat{q}_c^b(2:4) \end{bmatrix} \tag{3.6}$$

where $\delta\hat{q}_x^y = q_x^y \otimes (\hat{q}_x^y)^*$ and $\otimes$ is a quaternion multiplication.

These state mappings were simulated and verified using a MATLAB® simulation. In simulation, the values for the injected errors are known. This makes validation of the system of equations 3.4, 3.5, and 3.6 possible. Table 3.2 has the values of the error-corrected state vector subtracted from the state vector. Because the injected errors are known exactly in simulation, the error-corrected state (navigation state minus error state) should be exactly equal to the truth state. Because the simulation does not calculate exact numbers for quaternions, the error-corrected state for these is not always exact, but it is so close that it is only off by a factor of 10^{-16}, which is basically 0, indicating that the equations used to inject errors are consistent with the equations used to remove the errors.

The differences in Table 3.2 are considered to be sufficiently small, and the error state vector has been validated.

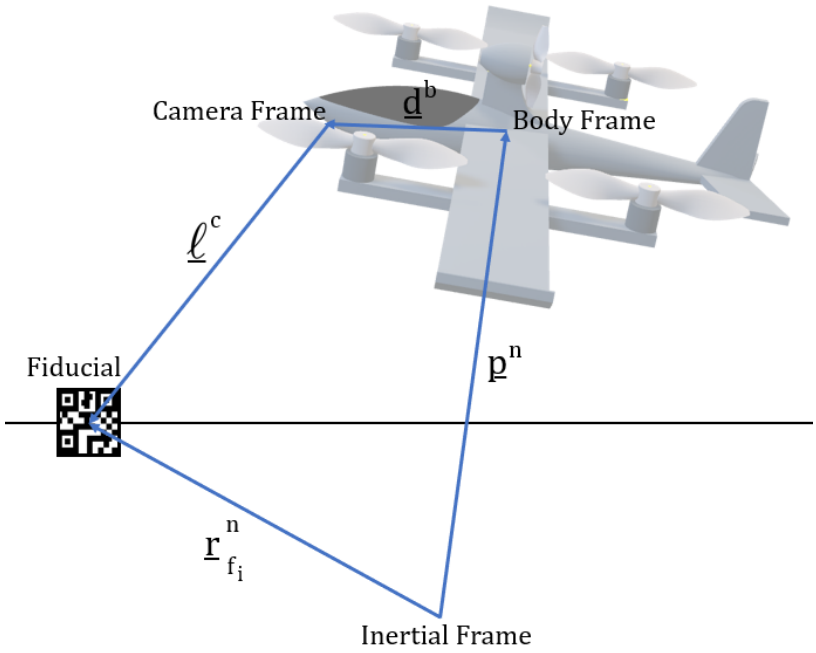

Fig. 3.2: Frames and relating vectors for state propagation

3.2 Nonlinear State Propagation and Measurement Modeling

Given the states defined in the previous section, the equations used to propagate those states are now defined. The IMU measurements that are used in the state propagation and state estimation are defined. Additional GPS and camera LOS measurements used in the state estimation are also defined. The state dynamics of the UAV are nonlinear, but a Kalman filter can only operate on a linear system. Therefore, the linearized dynamics of the design model are presented, and the nonlinear and linear models are compared and validated to be sufficiently similar.

To begin, it is useful to understand the vectors relating the location of the fiducial, the body and camera frames, and the inertial frame (shown in Figure 3.2) [9].

The relationships between the vectors in Figure 3.2 is also shown by Equation 3.7, where $\underline{\ell}^c$ is the LOS vector from the fiducial to the camera frame as expressed in the camera frame, $\underline{r}^n_{f_i}$ is the vector from the inertial frame to the ith fiducial as expressed in the NED frame, $\underline{p}^n$ is the position vector of the UAV as expressed in the NED frame, and

$\underline{d}^b$ is the displacement vector from the camera frame to the body frame.

$$\underline{\ell}^c = R_b^c R_n^b \underline{r}_{f_i}^n - R_b^c R_n^b \underline{p}^n - R_b^c \underline{d}^b \tag{3.7}$$

Other terms used in this section that are not previously defined in Table 3.1 are defined here in Table 3.3.

Term	Definition	Units
$\underline{\nu}^b$	Non-gravitational acceleration of the UAV as expressed in the body frame	m/s^2
$\underline{g}^n$	Gravity as expressed in the NED frame ($[0\ 0\ 9.8]^T$)	m/s^2
$\underline{\omega}^b$	Angular velocity of the UAV as expressed in the body frame	$radians/s$
τ_a	Time constant associated with the accelerometer.	s
$\underline{n}_a$	Accelerometer sensor noise	m/s^2
τ_g	Time constant associated with the gyroscope.	s
$\underline{n}_g$	Gyroscope sensor noise	$radians/s$
R_b^n	Rotation matrix for transformations from the body frame to the NED frame	unitless
R_b^c	Rotation matrix for transformations from the body frame to the camera frame	unitless
R_n^b	Rotation matrix for transformations from the NED frame to the body frame	unitless
z	Camera measurement model	$radians$
ℓ_x, ℓ_y, ℓ_z	X, Y, and Z components of the LOS vector between the camera and the fiducial as expressed in the camera frame	m

Table 3.3: Definition of additional terms for state propagation

The equations showing the propagation of the truth state ($\underline{\dot{x}}$) are shown in Equation

3.8. Note that many of the terms are the same as terms defined in Table 3.1, but here the dot above the term denotes the time derivative of that variable. For example, $\dot{\underline{p}}^n$ is the derivative of the position vector as expressed in the NED frame, which is the velocity vector as expressed in the NED frame.

The rate of change of position is simply equal to the velocity vector. The rate of change of the velocity vector is equal to the summation of true specific force (non-gravitational force acting on the UAV) summed with the force due to gravity. The rate of change of the attitude is directly related to the angular velocity of the UAV. The accelerometer and gyroscope biases are modeled as exponentially correlated random variables. The camera and body frames are fixed in reference to each other, so the quaternion representing the rotation from the camera frame to the body frame will have a derivative of the 0 vector.

$$\dot{\underline{x}} = \begin{bmatrix} \dot{\underline{p}}^n \\ \dot{\underline{v}}^n \\ \dot{q}_b^n \\ \dot{\underline{b}}_a \\ \dot{\underline{b}}_g \\ \dot{q}_c^b \end{bmatrix} = \begin{bmatrix} \underline{v}^n \\ R_b^n \, \underline{\nu}^b + \underline{g}^n \\ \frac{1}{2} \, q_b^n \otimes \begin{bmatrix} 0 \\ \underline{\omega}^b \end{bmatrix} \\ -\frac{1}{\tau_a} \, \underline{b}_a + \underline{n}_a \\ -\frac{1}{\tau_g} \, \underline{b}_g + \underline{n}_g \\ \begin{bmatrix} 0 \, 0 \, 0 \, 0 \end{bmatrix}^T \end{bmatrix} \tag{3.8}$$

Equation 3.9 shows the estimated state of the derivatives of the state vector. Note that $\tilde{}$ denotes a measured value. In this equation, accelerometer and gyroscope measurements are used to estimate states. These measurements are affected by the biases associated with them. Therefore the estimated rate of change in velocity is affected by the accelerometer bias, and the estimated rate of change in the attitude of the UAV is affected by the gyroscope

bias.

$$\dot{\hat{\underline{x}}} = \begin{bmatrix} \dot{\hat{\underline{p}}}^n \\ \dot{\hat{\underline{v}}}^n \\ \dot{\hat{q}}_b^n \\ \dot{\hat{\underline{b}}}_a \\ \dot{\hat{\underline{b}}}_g \\ \dot{\hat{q}}_c^b \end{bmatrix} = \begin{bmatrix} \hat{\underline{v}}^n \\ \hat{R}_b^n \left(\tilde{\underline{\nu}}^b - \hat{\underline{b}}_a\right) + \underline{g}^n \\ \frac{1}{2}\, q_{\hat{b}}^n \otimes \begin{bmatrix} 0 \\ \tilde{\underline{\omega}}^b - \hat{\underline{b}}_g \end{bmatrix} \\ -\frac{1}{\tau_a}\, \hat{\underline{b}}_a \\ -\frac{1}{\tau_g}\, \hat{\underline{b}}_g \\ \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T \end{bmatrix} \tag{3.9}$$

where

$$\tilde{\underline{\nu}}^b = \underline{\nu}^b + \underline{b}_a + \underline{n}_\nu \tag{3.10}$$

and

$$\tilde{\underline{\omega}}^b = \omega^b + b_g + n_\omega \tag{3.11}$$

In Equation 3.10, ν^b is the actual acceleration, b_a is the bias associated with the acceleration, and n_ν is the noise associated with the acceleration. In Equation 3.11, ω^b is the actual angular acceleration, b_g is bias associated with the angular acceleration, and n_ω is the noise associated with the angular acceleration.

When no noise is injected into the system, the difference between the truth state and estimated state derivatives should be 0. Equations 3.8 and 3.9 were implemented in a MATLAB simulation. The simulation used UAV flight data following the trajectory shown in Figure 3.3. This trajectory is also represented in a plot of position versus time in Figure 3.4. The trajectory begins at an altitude of 122.45 meters and proceeds into some S-curves. These S-curves help to exercise the dynamics of the UAV and provide a sort of calibration for the Kalman filter. After the S-curves, the UAV gradually descends and finishes at an

altitude of 15.38 meters. This descent is later used as a model for descending into a canyon where GPS measurements are unavailable.

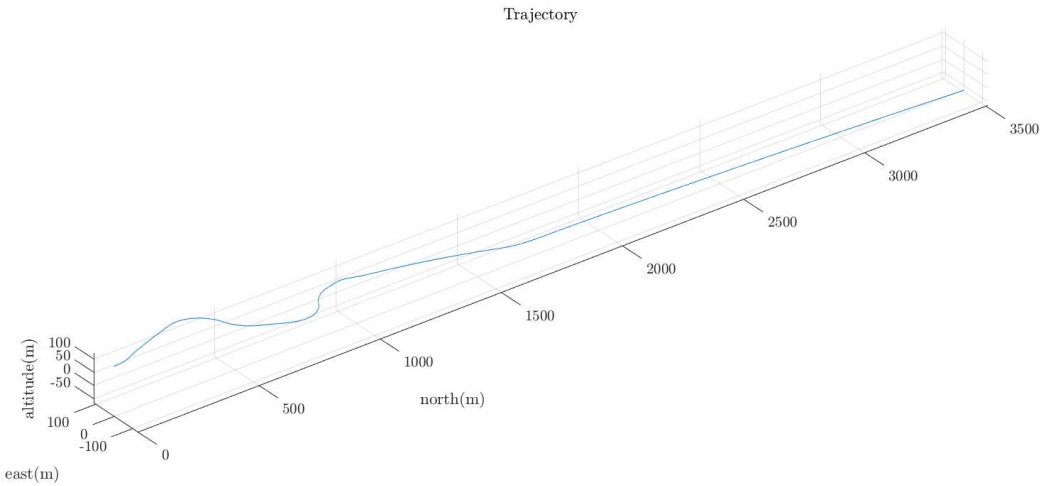

Fig. 3.3: Simulation UAV trajectory

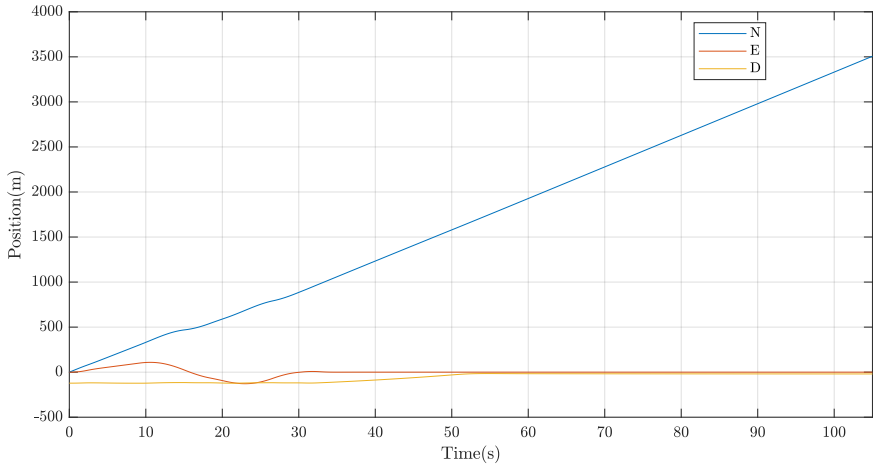

Fig. 3.4: Position of UAV over time

To verify that the truth state and navigation state differential equations were equivalent when no noise is present, the difference (error) between the two was plotted and verified to be 0. The plots of the error between the truth state and estimated state of the differential equations over the span of the simulated data are shown in Figures 3.5, 3.6, 3.7, and 3.8. These figures show that the difference between the propagation of the truth states and navigation states are equal.

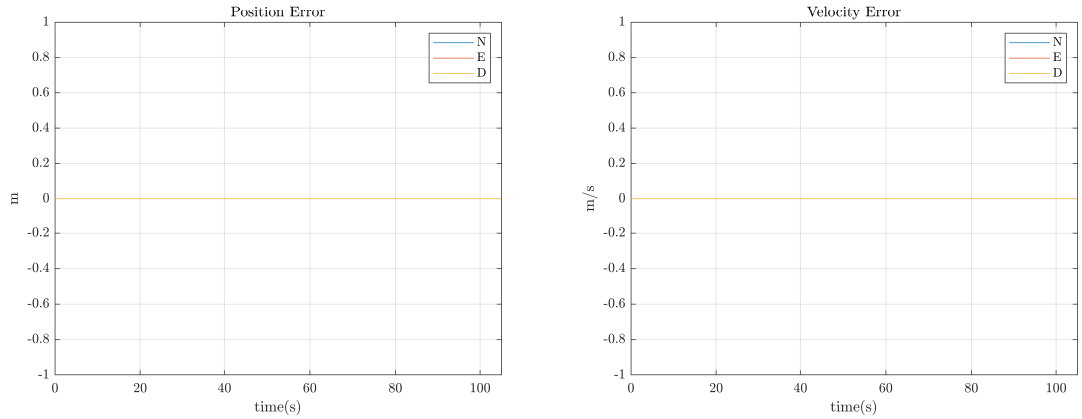

Fig. 3.5: Position error ($\underline{\dot{p}}^n - \dot{\underline{\hat{p}}}^n$) and velocity error ($\underline{\dot{v}}^n - \dot{\underline{\hat{v}}}^n$)

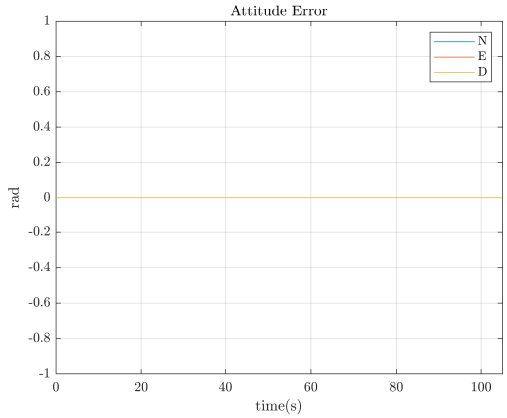

Fig. 3.6: UAV attitude error ($\dot{q}_b^n - \dot{\hat{q}}_b^n$)

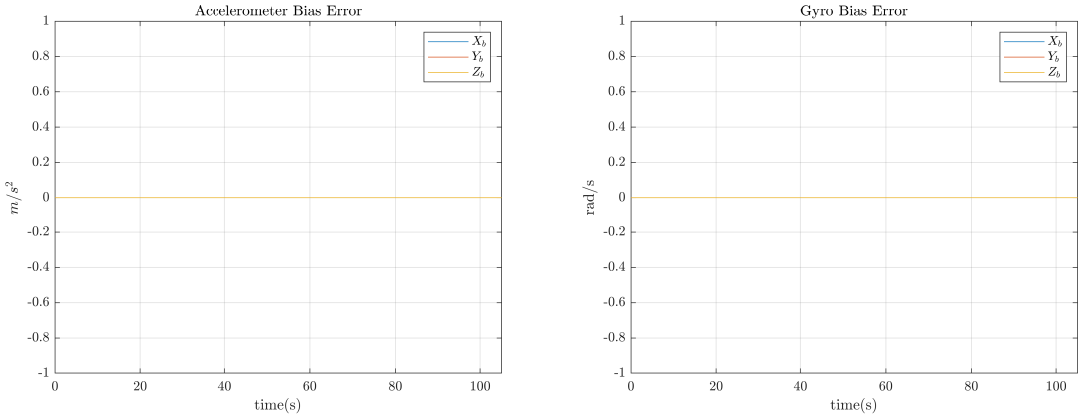

Fig. 3.7: Accelerometer bias error ($\dot{\underline{b}}_a - \dot{\hat{\underline{b}}}_a$) and gyro bias error ($\dot{\underline{b}}_g - \dot{\hat{\underline{b}}}_g$)

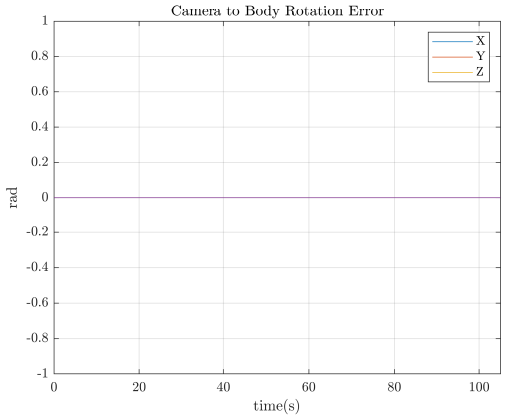

Fig. 3.8: Camera-to-body frame rotation error ($\dot{q}_c^b - \hat{\dot{q}}_c^b$)

These residuals are the difference between the truth state and navigation state propagated using IMU measurements without any noise added. If the propagation is done properly, these residuals should be 0. The fact that the residuals are 0 throughout the simulation indicates that both the truth state and navigation states are propagated properly.

Equations 3.12 and 3.13 show the calculation of the LOS truth and estimated state vectors between the camera and the fiducial as expressed in the camera frame. First, the vector from the inertial frame to the NED frame is transformed to be expressed in the camera frame. The position of the UAV as estimated using the propagation equations is also transformed to be expressed in the camera frame and then subtracted from the fiducial's position vector to get the LOS vector. Figure 3.2 is a good visualization of this.

$$\underline{\ell}^c = R_b^c \, R_n^b \, \underline{r}_{f_i}^n - R_b^c \, R_n^b \, \underline{p}^n - R_b^c \, \underline{d}^b \tag{3.12}$$

$$\hat{\underline{\ell}}^c = \hat{R}_b^c \, \hat{R}_n^b \, \underline{r}_{f_i}^n - \hat{R}_b^c \, \hat{R}_n^b \, \hat{\underline{p}}^n - \hat{R}_b^c \, \underline{d}^b \tag{3.13}$$

The camera gets a two-dimensional picture of the location of the fiducial, so the x, y, and z elements of the fiducial's position vector are translated into an x and y two-element vector $\underline{z}$ as shown in equations 3.14 and 3.15. For this verification step of the simulation,

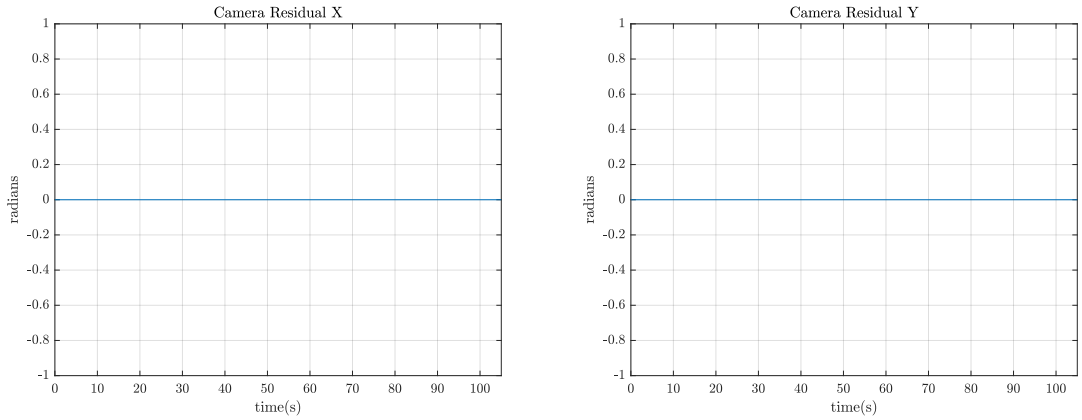

Fig. 3.9: Camera measurement error ($\underline{\tilde{z}} - \underline{\hat{\tilde{z}}}$)

ν_ℓ (the camera noise in its interpretation of the fiducial's location) was set to 0.

$$\underline{\tilde{z}} = \begin{bmatrix} \frac{\ell_x}{\ell_z} \\ \\ \frac{\ell_y}{\ell_z} \end{bmatrix} + \underline{\nu}_\ell \tag{3.14}$$

$$\underline{\hat{\tilde{z}}} = \begin{bmatrix} \frac{\hat{\ell}_x}{\hat{\ell}_z} \\ \\ \frac{\hat{\ell}_y}{\hat{\ell}_z} \end{bmatrix} \tag{3.15}$$

Figure 3.9 shows the residual of the difference between the truth state and the estimated state of the camera measurement model $\underline{z}$.

The GPS measurement is modeled by the position of the UAV in the navigation frame corrupted by zero-mean, white, Gaussian noise.

$$\underline{\tilde{z}}_{gps} = \underline{p}^n + \underline{\eta} \tag{3.16}$$

The estimated GPS measurement is simply the estimated position as shown in Equation 3.17.

$$\underline{\hat{\tilde{z}}}_{gps} = \underline{\hat{p}}^n \tag{3.17}$$

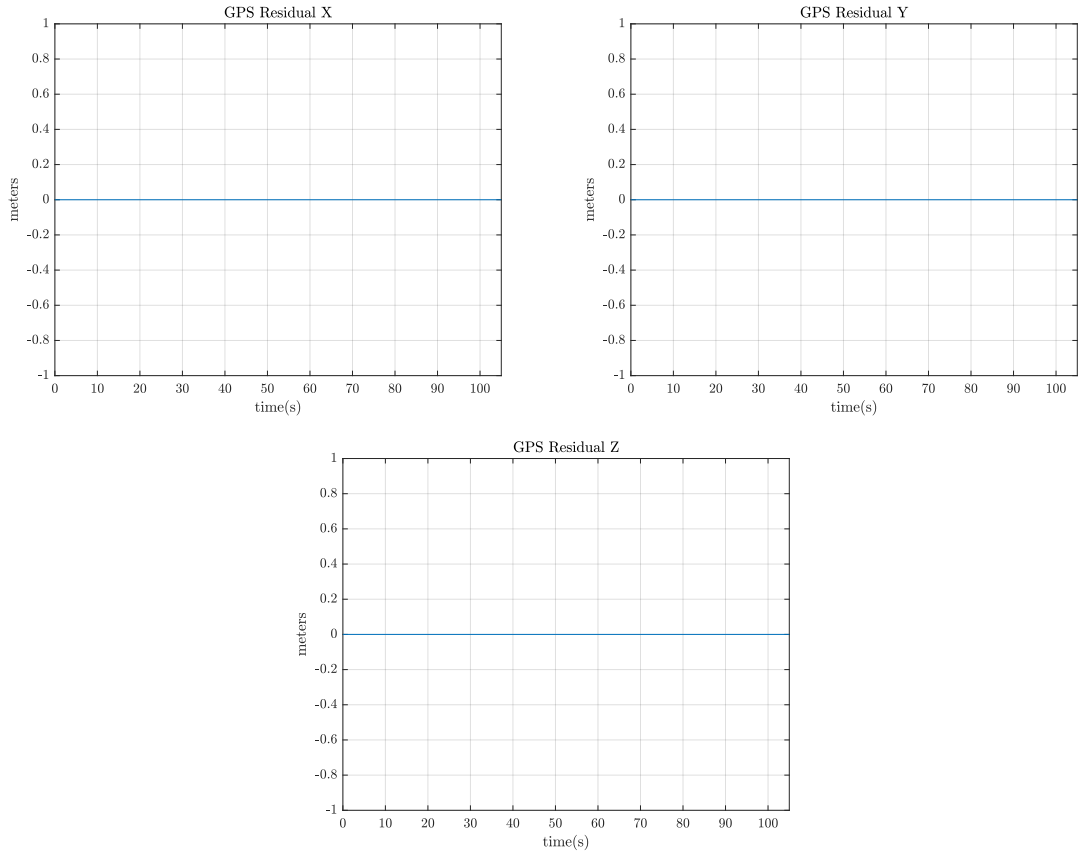

Fig. 3.10: GPS measurement error ($\tilde{\underline{z}}_g ps - \hat{\tilde{\underline{z}}}_g ps$)

Figure 3.10 shows the residual of the difference between the truth state and the estimated state of the GPS measurement model $\underline{z}_g ps$.

Both the camera measurement residuals and the GPS measurement residuals are 0 throughout the simulation. This indicates that the equations used to inject and remove error from the camera measurements were implemented correctly.

3.3 Linear Error State Modeling

Now that the nonlinear equations have been validated, linear equations need to be developed and validated because a Kalman filter can only operate on linear equations. The design model dynamics of the UAV (Equation 3.8) were linearized about the state estimates $\hat{\underline{x}}$. Appendix A shows the derivation of the linearized state equations, the result of which is shown here in Equation 3.18. This section shows the resulting differences between

propagating the states of the UAV with the nonlinear equations and linear equations to show that the difference is sufficiently small. This shows that the linear equations are valid approximations of the nonlinear equations and can be used in the Kalman filter.

$$\delta\dot{\underline{x}} = \begin{bmatrix} \delta\dot{\underline{p}}^n \\ \delta\dot{\underline{v}}^n \\ \delta\dot{\underline{\theta}}_b^n \\ \delta\dot{\underline{b}}_a \\ \delta\dot{\underline{b}}_g \\ \delta\dot{\underline{\theta}}_c^b \end{bmatrix} = \begin{bmatrix} \delta\underline{v}^n \\ [\hat{R}_b^n(\tilde{\underline{\nu}}^b - \hat{\underline{b}}_a)] \times \delta\underline{\theta}_b^n - \hat{R}_b^n\delta\underline{b}_a - \hat{R}_b^n\underline{n}_\nu \\ \hat{R}_b^n\delta\underline{b}_g + \hat{R}_b^n\underline{n}_\omega \\ -\frac{1}{\tau_a}\delta\underline{b}_a + \underline{n}_a \\ -\frac{1}{\tau_g}\delta\underline{b}_g + \underline{n}_g \\ \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \end{bmatrix} \tag{3.18}$$

Equation 3.18 can also be expressed in the form of Equation 3.19 where F is expressed in Equation 3.20, B is expressed in Equation 3.21, and $\underline{w}$ is expressed in Equation 3.22. This section validates that the error state dynamics matrix F developed here is implemented correctly.

$$\delta\dot{\underline{x}} = F\left(\hat{\underline{x}}\right)\delta\underline{x} + B\underline{w} \tag{3.19}$$

$$F = \begin{bmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & \left(\left(\hat{R}_b^n\left(\tilde{\underline{\nu}}^b - \hat{\underline{b}}_a\right)\right)\times\right) & -\hat{R}_b^n & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \hat{R}_b^n & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \tau_a & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \tau_g & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{3.20}$$

$$B = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ -\hat{R}_b^n & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \hat{R}_b^n & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{3.21}$$

$$\underline{w} = \begin{bmatrix} \underline{n}_\nu \\ \underline{n}_\omega \\ \underline{n}_a \\ \underline{n}_g \end{bmatrix} \tag{3.22}$$

The error state $\delta\underline{x}$ resulting from integrating Equation 3.18 was compared to the differences computed in Section 3.2 after one propagation cycle in the MATLAB simulation. The difference between $\delta\underline{x}$ from the linear and nonlinear equations is shown in the last column of Table 3.4.

	$\delta\underline{x}$ Linear	$\delta\underline{x}$ Nonlinear	Propagation Error
$\delta\underline{p}^n_x$	0.213317	0.213308	8.954259e-6
$\delta\underline{p}^n_y$	0.384864	0.384824	3.989025e-5
$\delta\underline{p}^n_z$	0.585298	0.585328	-2.954861e-5
$\delta\underline{v}^n_x$	0.126675	0.126658	1.790756e-5
$\delta\underline{v}^n_y$	0.169765	0.169686	7.974998e-5
$\delta\underline{v}^n_z$	0.270678	0.270737	-5.908257e-5
$\delta\underline{\theta}^n_{b,x}$	0.000997	0.000997	9.414459e-10
$\delta\underline{\theta}^n_{b,y}$	0.002010	0.002010	-8.912999e-9
$\delta\underline{\theta}^n_{b,z}$	0.003014	0.003014	1.380391e-8
$\delta\underline{b}_a, x$	0.009648	0.009648	0
$\delta\underline{b}_a, y$	0.019296	0.019296	0
$\delta\underline{b}_a, z$	0.028944	0.028944	0
$\delta\underline{b}_g, x$	4.768004e-6	4.768004e-6	0
$\delta\underline{b}_g, y$	9.536008e-6	9.536008e-6	0
$\delta\underline{b}_g, z$	1.430401e-5	1.430401e-5	0
$\delta\underline{\theta}^b_{c,x}$	0.001	0.001	1.749995e-9
$\delta\underline{\theta}^b_{c,y}$	0.002	0.002	3.499991e-9
$\delta\underline{\theta}^b_{c,z}$	0.003	0.003	5.249986e-9

Table 3.4: $\delta\underline{x}$ in meters calculated using the linear and nonlinear equations and the propagation error difference

The errors in Table 3.4 are of the magnitude of 10^{-5} at the largest. The largest error is in the sub-millimeter range where the corresponding $\delta\underline{x}$ value is in the tenth of a meter range. These errors are considered to be sufficiently small for UAV navigation applications discussed in this paper. These linearized equations can now be used to propagate both the mean and covariance of the state vector, and the F matrix is considered to be validated. The B matrix was not yet validated in this section because all noise was turned off.

3.4 Linear Measurement Modeling

This section discusses the linearization of the GPS measurements as well as the camera measurements. Small estimation errors are injected into both the nonlinear models and the linear models, and the measurements using both models are compared. The difference is verified to be sufficiently small.

Recall the truth (and design) model of the LOS measurement from Equation 3.14:

$$\underline{\tilde{z}} = \begin{bmatrix} \frac{\ell_x}{\ell_z} \\ \\ \frac{\ell_y}{\ell_z} \end{bmatrix} + \underline{\nu}_\ell$$

The camera obtains images in two dimensions, so it only has two measurements for the location of the fiducials: x and y. In the three-dimensional world in which the fiducials reside, there are three measurements: x, y, and z. The further an object is away, the smaller it appears in an image. The space an object takes up in an image is a scaled version of how much space it takes up in physical space. The closer the object is to the camera, the larger it appears in the image. Therefore, x and y values corresponding to a point in an image are a combination of the x, y, and z components of the point in three-dimensional space. For a single point located at position (x, y, z) in three-dimensional space, the point will appear at position (x/z, y/z) multiplied by a scale factor in an image. This is true because of the characteristics of similar triangles. These ratios between the x, y, and z distances are what is stored in the z vector. The noise associated with this camera measurement is $\underline{\nu}_\ell$.

The measurement model for the LOS was linearized about the current estimate $\underline{\hat{\tilde{z}}}$. Recall the nominal LOS measurement from Equation 3.15:

$$\underline{\hat{\tilde{z}}} = \begin{bmatrix} \frac{\hat{\ell}_x}{\hat{\ell}_z} \\ \\ \frac{\hat{\ell}_y}{\hat{\ell}_z} \end{bmatrix}$$

The linearization process for the LOS measurement can be seen in Appendix B. The linearized measurement model is shown in Equation 3.23 where the sensitivity matrices

$H_{\delta p^n}$, $H_{\delta\theta_b^n}$, and $H_{\delta\theta_c^b}$ are shown in equations 3.24, 3.25, and 3.26. These H matrices are validated in this section.

$$\delta\underline{\ell}^c = H_{\delta p^n}\delta\underline{p}^n + H_{\delta\theta_b^n}\delta\underline{\theta}_b^n + H_{\delta\theta_c^b}\delta\underline{\theta}_c^b + \underline{\nu}_\ell \tag{3.23}$$

$$H_{\delta p^n} = -\hat{R}_b^c\hat{R}_n^b \tag{3.24}$$

$$H_{\delta\theta_b^n} = \hat{R}_b^c\hat{R}_n^b\left[(\hat{\underline{p}}^n - \underline{r}_{f_i}^n)\times\right] \tag{3.25}$$

$$H_{\delta\theta_c^b} = -\hat{R}_b^c\left\{\left[\hat{R}_n^b\left(\underline{r}_{f_i}^n - \hat{\underline{p}}^n\right) - \underline{d}^b\right]\times\right\} \tag{3.26}$$

The residual $\delta\underline{z}$ associated with the LOS measurement is shown in Equation 3.27.

$$\delta\underline{z} = \tilde{\underline{z}} - \hat{\tilde{\underline{z}}} \tag{3.27}$$

This residual is the difference between the actual LOS measurements and the measurements estimated using the navigation states. These residual values were compared to the values of $\delta\underline{\ell}^c$ obtained through the linearized Equation 3.23 in a MATLAB simulation. Small estimation errors were injected into both the linear models as shown in Equations 3.23 and 3.28 and the nonlinear model as shown in Equations 3.5. The injected errors can be seen in Table 3.5.

State	$\delta\underline{p}^n$ (m)	$\delta\underline{v}^n$ ($\frac{m}{s}$)	$\delta\underline{\theta}_b^n$ (rad)	$\delta\underline{b}_a$ (g)	$\delta\underline{b}_g$ ($\frac{deg}{hr}$)	$\delta\underline{\theta}_c^b$ (rad)
X Injected Error	0.1	0.1	0.001	0.0001	1	0.001
Y Injected Error	0.2	0.2	0.002	0.0002	2	0.002
Z Injected Error	0.3	0.3	0.003	0.0003	3	0.003

Table 3.5: Errors injected during MATLAB simulation (Note that g is the acceleration of gravity)

	$\delta\underline{z}$ nonlinear	$\delta\underline{z}$ linear	difference
X component (rad)	0.0026394	0.0026023	3.7092e-5
Y component (rad)	-0.011886	-0.011748	-0.0001381

Table 3.6: Difference between residual as calculated using the nonlinear equations and the residual as calculated using $H * \delta\underline{x}$

The values obtained for both the residual value $\delta\underline{z}$ obtained through the nonlinear equations and the value of $\delta\underline{\ell}^c$ obtained through the linear equations is shown in Table 3.6 along with the difference between the two.

The difference between the results obtained using the nonlinear and linear equations is considered to be sufficiently small, and the nonlinear equations are considered to be sufficient estimates of the model. This validates correct implementation of the H matrix.

The GPS measurement is already linear, so going from the measurement to the linear perturbation is trivial. It is shown in Equation 3.28.

$$\delta\underline{z}_{gps} = \delta\underline{p}^n + \underline{\eta} \tag{3.28}$$

The GPS measurement residual is shown in Equation 3.29.

$$\delta\underline{z}_{gps} = \tilde{\underline{z}}_{gps} - \hat{\tilde{\underline{z}}}_{gps} \tag{3.29}$$

The values obtained for both the residual value $\delta\underline{z}_{gps}$ obtained through the nonlinear equations and the value of $\delta\underline{z}_{gps}$ obtained through the linear equations is shown in Table 3.7 along with the difference between the two.

	$\delta\underline{z}$ nonlinear	$\delta\underline{z}$ linear	difference
X component (m)	0.100000000000001	0.1	1.416e-15
Y component (m)	0.2	0.2	1.665e-16
Z component (m)	0.299999999999997	0.3	-2.831e-15

Table 3.7: Difference between X, Y, and Z components of the nonlinear and linear GPS measurement modeling

The difference between the linear and nonlinear measurement models is considered to be sufficiently small, and the nonlinear models for the GPS and camera measurements are now able to be used in the Kalman filter. The B and H matrices are also considered to be validated.

3.5 Covariance Propagation

In this section, errors are injected into the IMU measurements so that the estimated state propagation no longer exactly matches the truth state propagation. Monte Carlo simulations are run to verify that as the state propagation is done many times, the resulting errors are zero-mean and stay mostly within 3-σ bounds. The F matrix has been validated in Section 3.3. This section validates the error state dynamics matrix B and process noise strength matrix Q.

The propagation of the covariance of the estimation errors can be calculated using Equation 3.30.

$$\dot{P} = F\left(\hat{\underline{x}}\right)P + PF^{T}\left(\hat{\underline{x}}\right) + BQB^{T} \tag{3.30}$$

The F matrix in Equation 3.30 is as expressed in Equation 3.20, and the process noise coupling matrix B is as expressed in Equation 3.21. The elements of the process noise vector $\underline{w}$ as defined in Equation 3.22 are independent, zero-mean, white, Gaussian noise sources. The power spectral density of the noise vector $\underline{w}$ is shown in Equation 3.31.

$$E\left[\underline{w}\left(t\right)\underline{w}^{T}\left(t'\right)\right] = Q\delta\left(t - t'\right) \tag{3.31}$$

The noise strength Q of the noise vector $\underline{w}$ (which was defined in Equation 3.22) is shown in Equation 3.32 with the individual noise strengths of each noise source on the

diagonal and zeros on the off-diagonal.

$$Q = \begin{bmatrix} Q_{\underline{n}_\nu} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & Q_{\underline{n}_\omega} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & Q_{\underline{n}_a} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & Q_{\underline{n}_g} \end{bmatrix} \tag{3.32}$$

The individual noise strengths $Q_{\underline{n}_\nu}$, $Q_{\underline{n}_\omega}$, $Q_{\underline{n}_a}$, and $Q_{\underline{n}_g}$ are shown in equations 3.33 to 3.36 where $\sigma_{ss,a}$ is the steady-state standard deviation associated with the accelerometer measurements from the Inertial Measurement Unit (IMU), and $\sigma_{ss,g}$ is the steady-state standard deviation associated with the gyroscope measurements from the IMU.

$$Q_{\underline{n}_\nu} = \begin{bmatrix} VRW^2 & 0 & 0 \\ 0 & VRW^2 & 0 \\ 0 & 0 & VRW^2 \end{bmatrix} \tag{3.33}$$

$$Q_{\underline{n}_\omega} = \begin{bmatrix} ARW^2 & 0 & 0 \\ 0 & ARW^2 & 0 \\ 0 & 0 & ARW^2 \end{bmatrix} \tag{3.34}$$

$$Q_{\underline{n}_a} = \frac{2\sigma_{ss,a}^2}{\tau_a} I_{3\times3} \tag{3.35}$$

$$Q_{\underline{n}_g} = \frac{2\sigma_{ss,g}^2}{\tau_g} I_{3\times3} \tag{3.36}$$

A MATLAB simulation of the error state variance was run using the IMU parameters displayed in Table 3.8, where VRW is Velocity Random Walk, and ARW is Angle Random Walk.

The initial 3-σ values of the states were set to the values shown in Table 3.9.

Parameter	$\sigma_{ss,a}$	τ_a	VRW	$\sigma_{ss,g}$	τ_g	ARW
Value	0.001 g	10 s	0.06 $m/s/\sqrt{hr}$	1.0 deg/hr	10 s	0.07 $deg/\sqrt{hr}$

Table 3.8: IMU specification 3-σ values

	Initial 3-σ Values		
State	X	Y	Z
Position	1.0 m	1.0 m	3.0 m
Velocity	0.1 m/s	0.1 m/s	0.1 m/s
Attitude	0.1 rad	0.1 rad	0.1 rad
Camera Orientation	0.1 rad	0.1 rad	0.1 rad
Accelerometer Bias	0.001 g	0.001 g	0.001 g
Gyroscope Bias	1.0 deg/hr	1.0 deg/hr	1.0 deg/hr

Table 3.9: Initial Uncertainty 3-σ values

The estimated error in the states of the UAV was plotted for each of 200 simulations as hairlines in Figures 3.11 to 3.16. The 3-σ values are also propagated using Equation 3.30 and plotted to show the bounds within which 99.7% of the data should lie. Each of the states should be zero-mean, so the plots should be centered around zero as well as be within the 3-σ lines. Visual inspection shows this to be true for most of the plots. However, a bias is accrued that shifts both the position and the velocity states in the down direction.

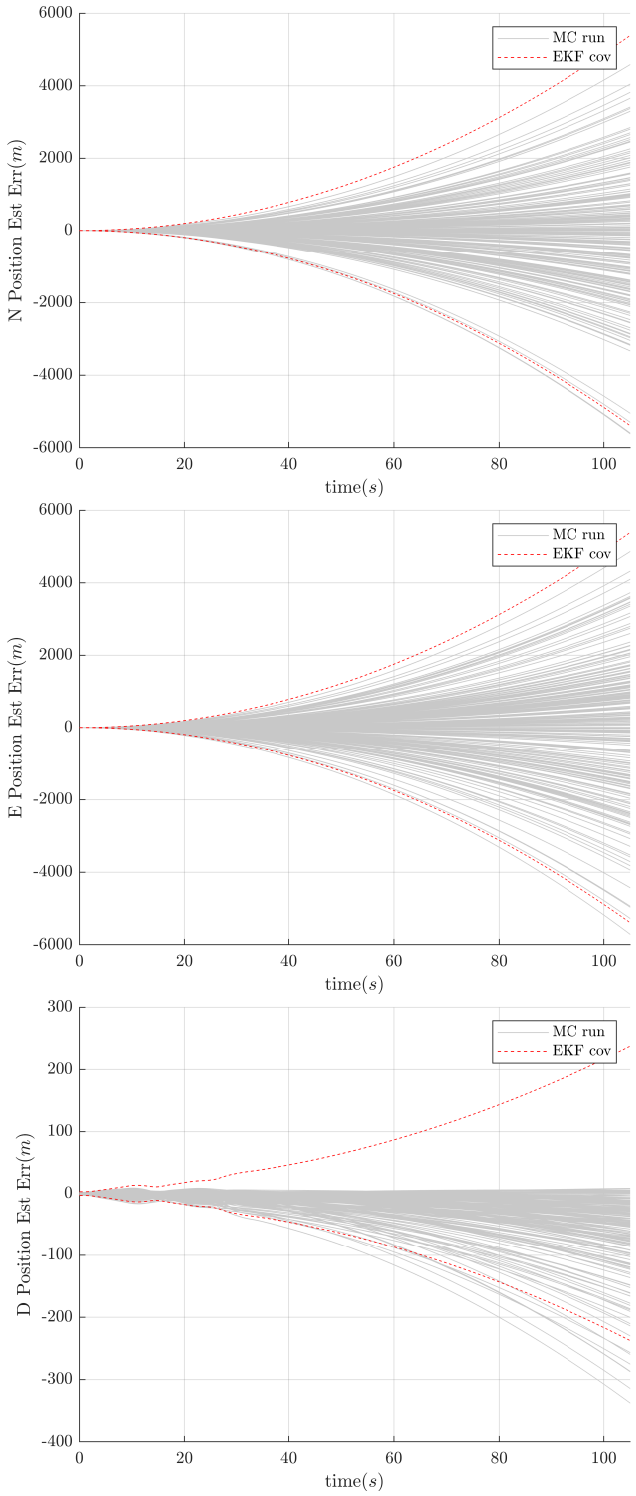

Fig. 3.11: North, east, and down position estimation error

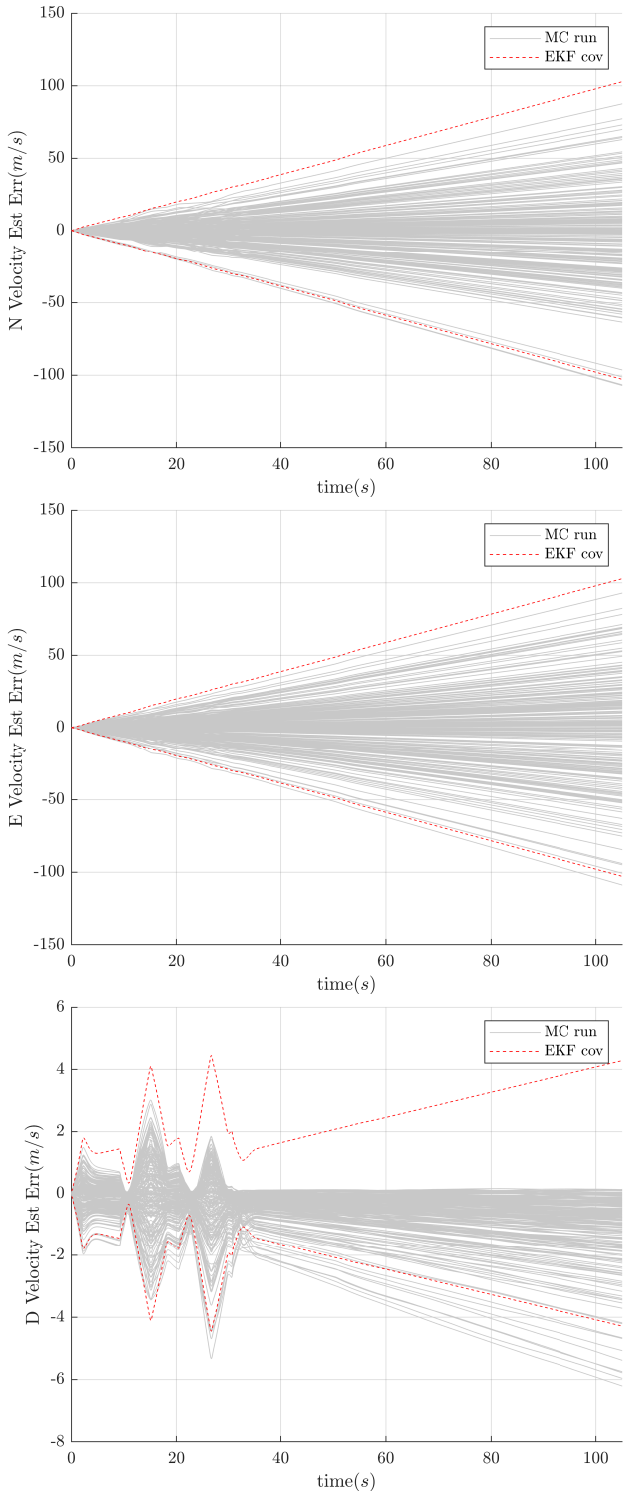

Fig. 3.12: North, east, and down velocity estimation error

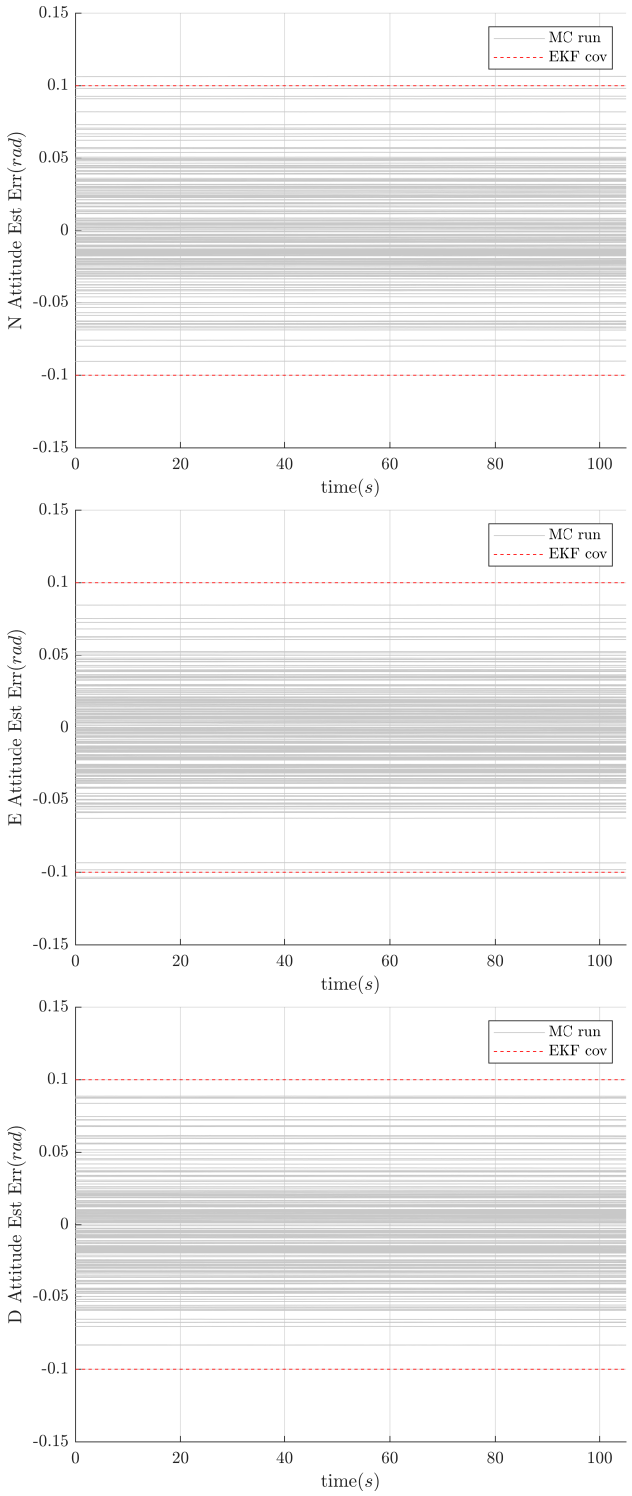

Fig. 3.13: North, east, and down attitude estimation error

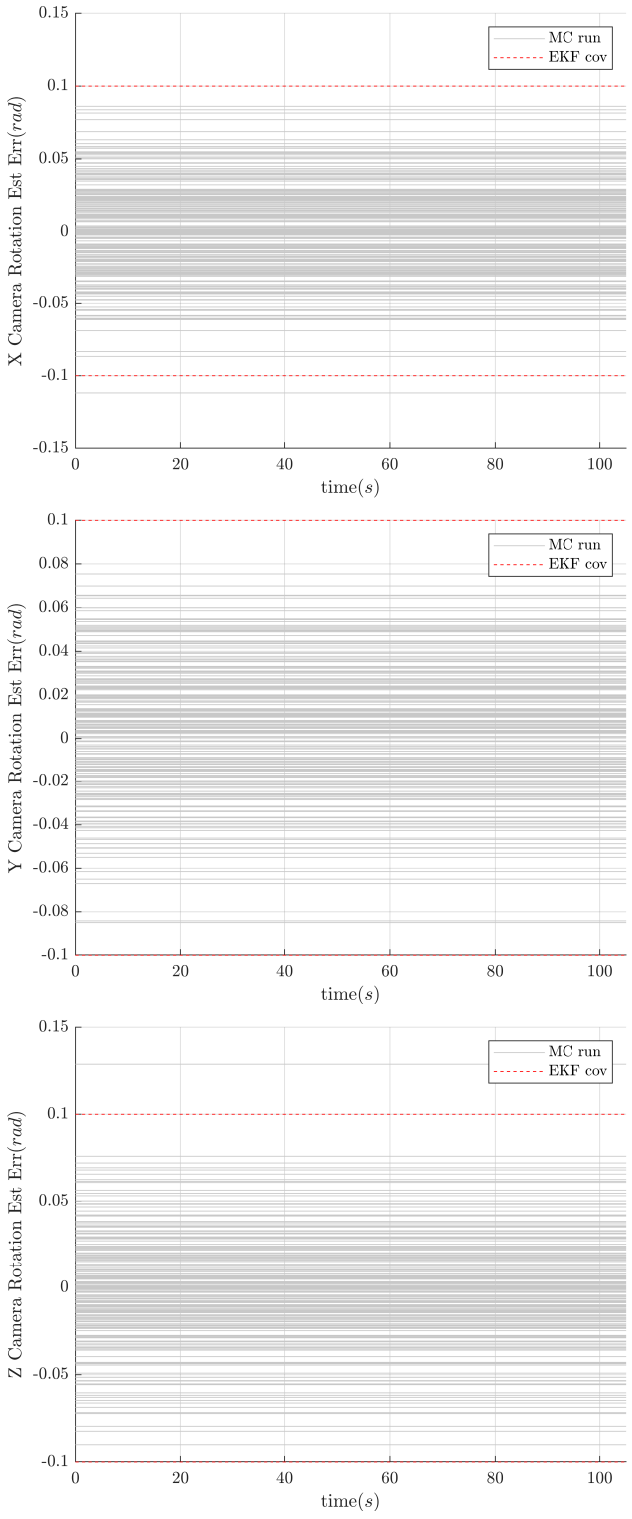

Fig. 3.14: X, Y, and Z camera attitude estimation error

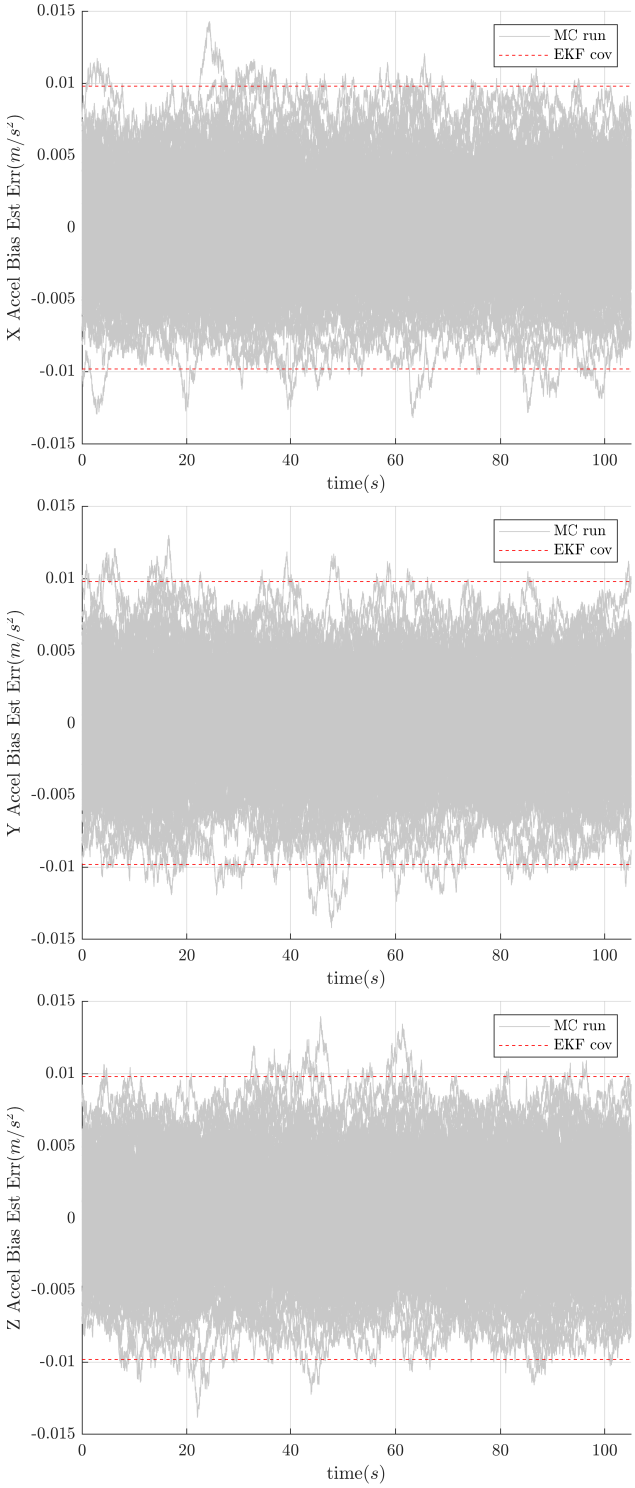

Fig. 3.15: X, Y, and Z accelerometer bias estimation error

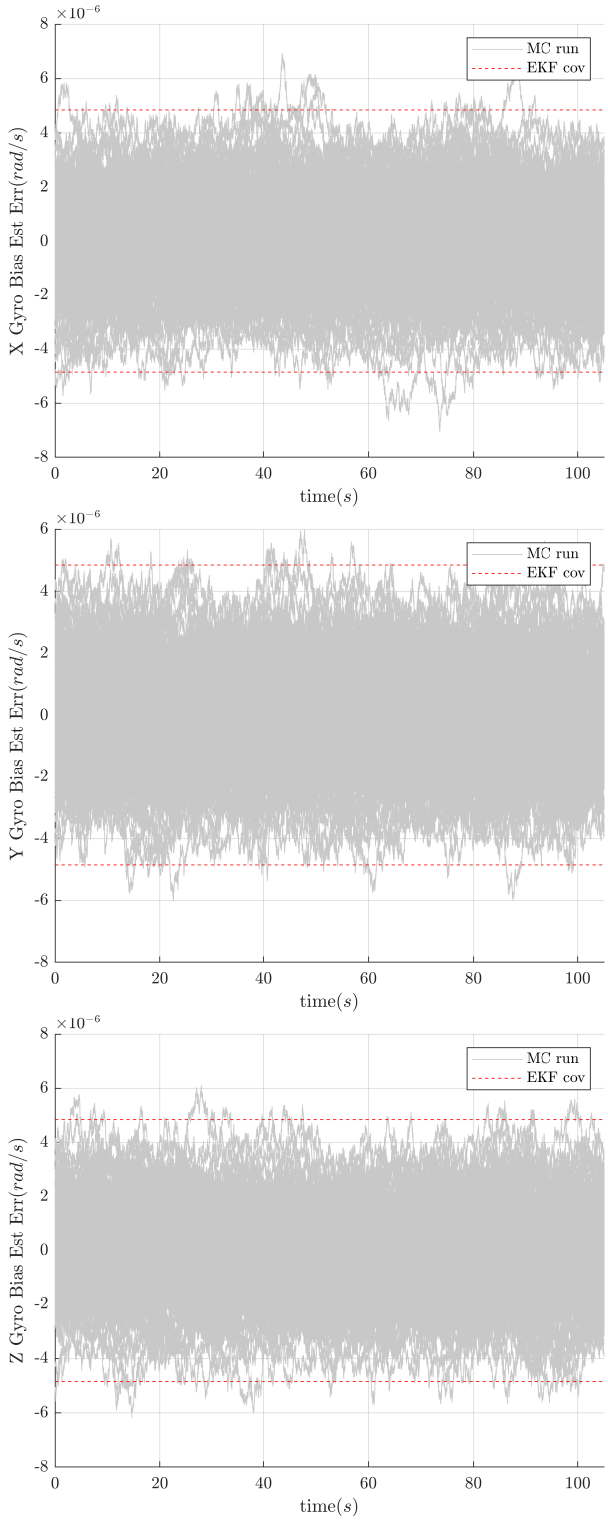

Fig. 3.16: X, Y, and Z gyro bias estimation error

During the attempt to find the source of the downward bias in both the position and velocity estimate errors, the initial uncertainties of each of the states were set to zero one-by-one. It was found that setting the initial uncertainty for the attitude of the UAV had the most effect on the unwanted downward bias. It was further found that reducing the initial uncertainty in the attitude by a factor of 10 to 0.001 radians in the X, Y, and Z directions removed the downward bias. The plots resulting from this adjustment can be seen in Figures 3.17 through 3.22.

This connection between the attitude uncertainty and the downward bias in the down direction of the velocity and position can be accounted for by the fact that this system is a nonlinear system being approximated with linear equations. When the attitude estimation is inaccurate, the second-order terms have enough effect to bias the estimates. This bias shifts errors downward in the plots for the down direction of the velocity and position. Gravity is applied in a constant down direction in the NED frame as the EKF estimates the states of the UAV. When there is error in the attitude of the UAV, the gravity acceleration vector is applied incorrectly to the UAV dynamics, always in the down direction. Without any measurements to provide absolute updates to the state estimates, over time, this downward pull manifests as a downward bias in the estimation errors.

For the remainder of the simulation portion of this study, the 3-σ initial uncertainty of the attitude estimates is set to 0.01 radians. All of the estimation errors in Figures 3.17 through 3.22 appear to be zero-mean, and the estimation errors stay sufficiently within the 3-σ bounds, so the covariance propagation is considered to be implemented correctly, and the B and Q matrices have been validated.

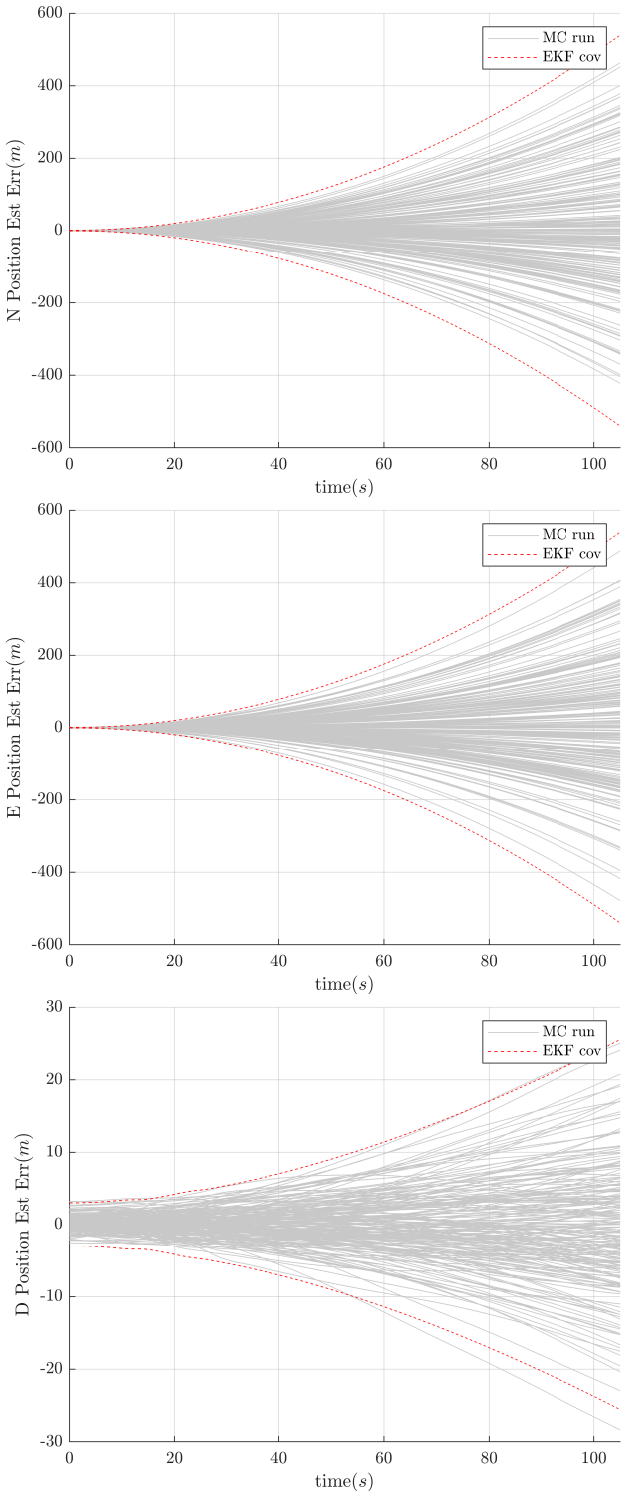

Fig. 3.17: North, east, and down position estimation error after initial attitude uncertainty adjustment

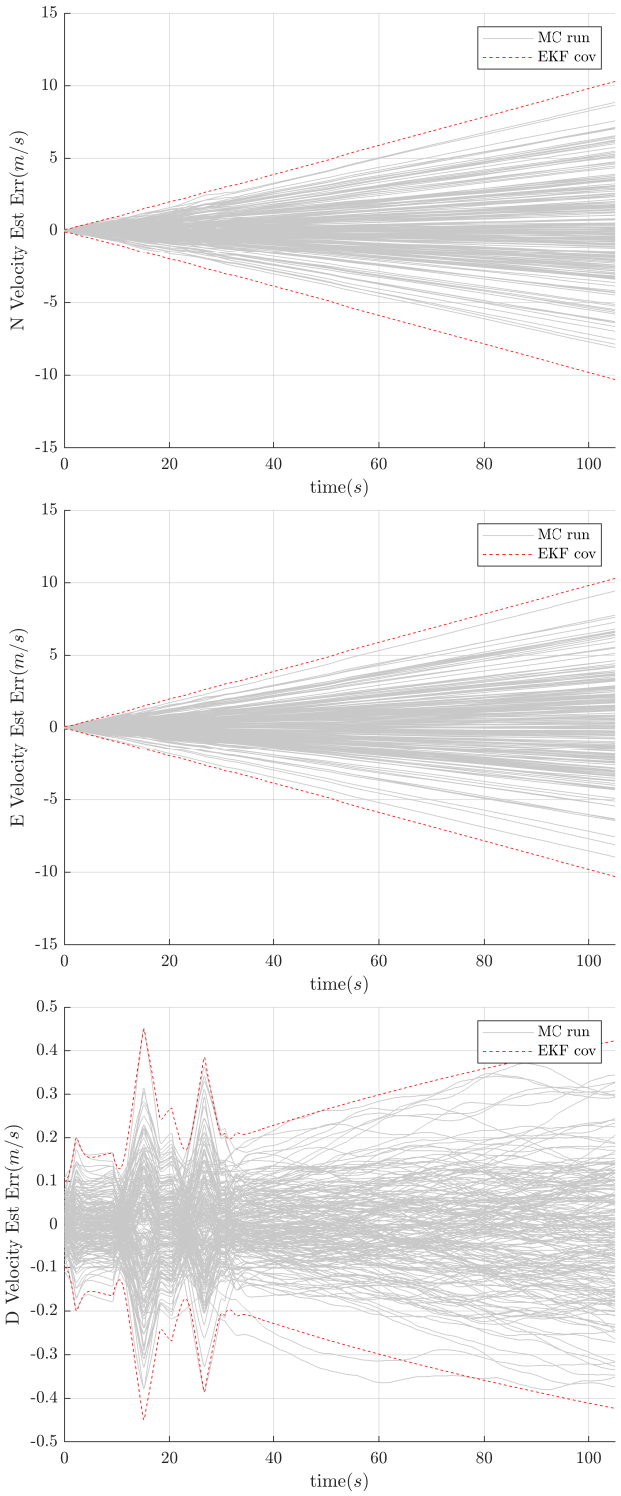

Fig. 3.18: North, east, and down velocity estimation error after initial attitude uncertainty adjustment

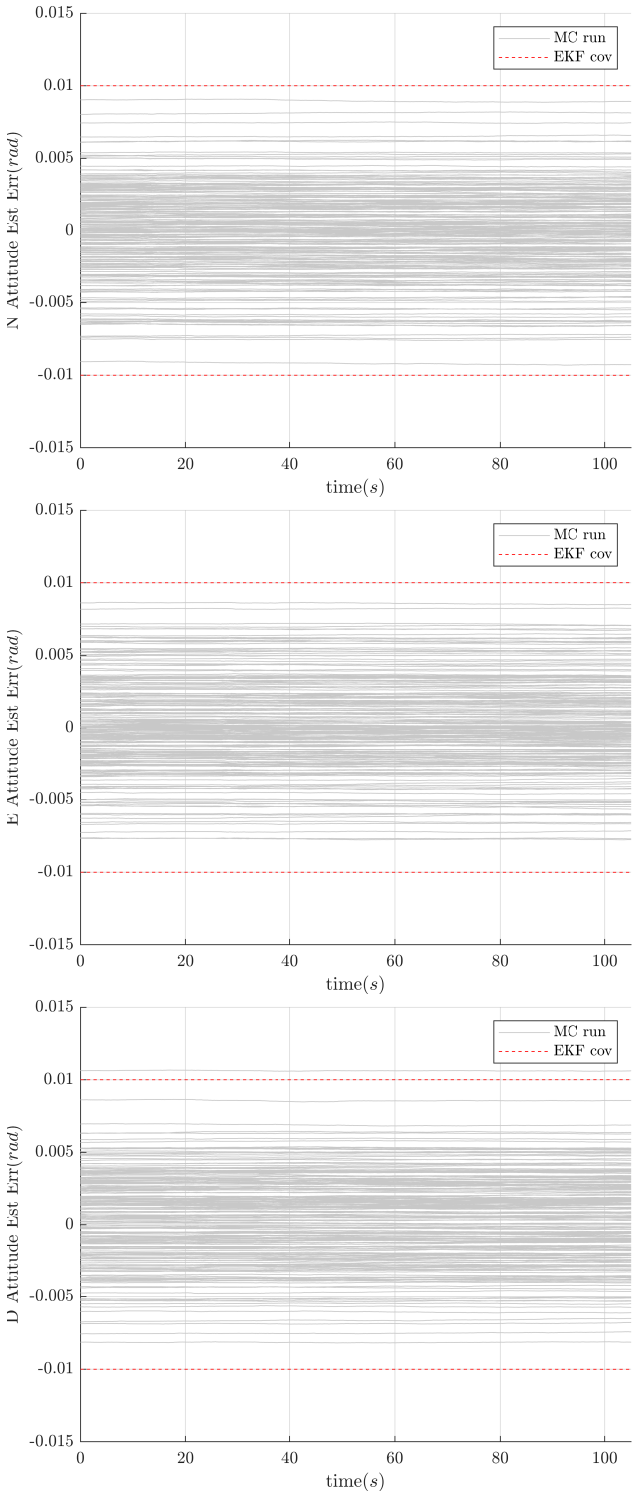

Fig. 3.19: North, east, and down attitude estimation error after initial attitude uncertainty adjustment

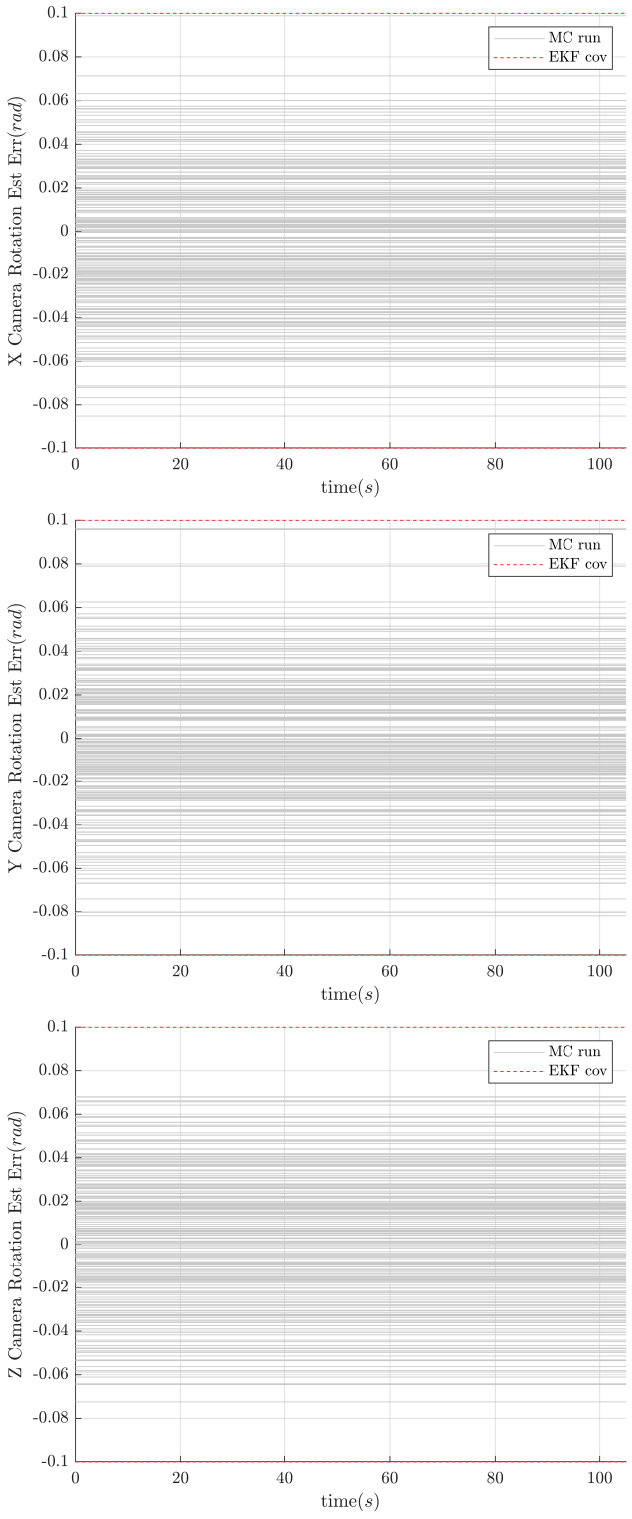

Fig. 3.20: X, Y, and Z camera attitude estimation error after initial attitude uncertainty adjustment

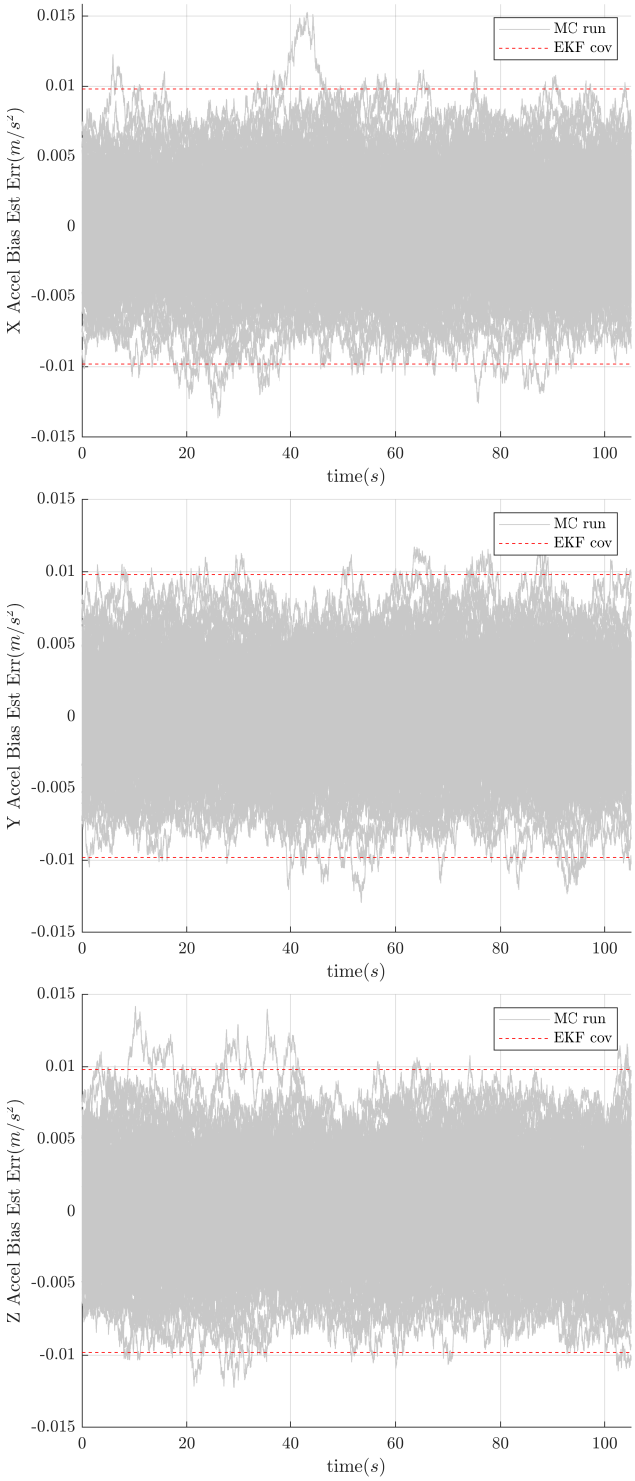

Fig. 3.21: X, Y, and Z accelerometer bias estimation error after initial attitude uncertainty adjustment

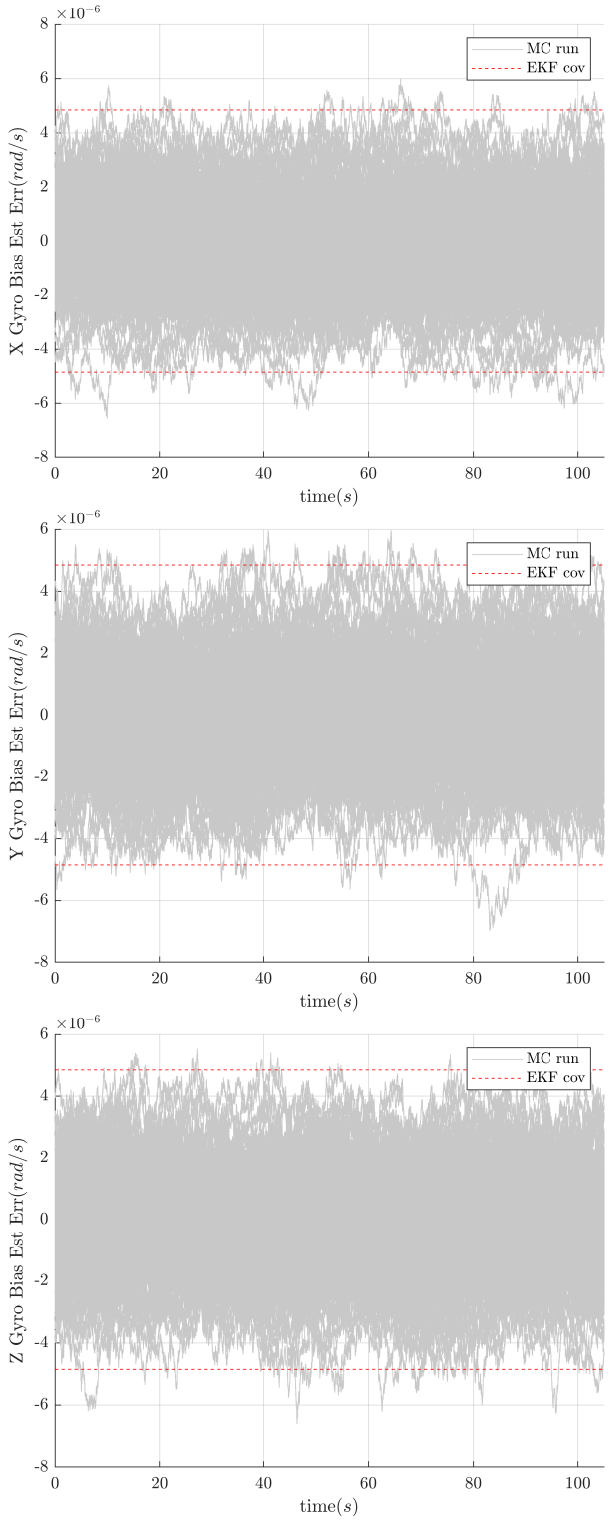

Fig. 3.22: X, Y, and Z gyro bias estimation error after initial attitude uncertainty adjustment

3.6 Estimation Capability

Using the components validated in previous sections, the Kalman update equations were implemented in this section. This section validates that the state estimation error covariance is updated properly once measurements are added into the Kalman filter and that this covariance is reflected as 200 Monte Carlo simulations of the EKF are run with both GPS and camera measurement updates implemented. This validates the measurement noise coupling matrix G as well as the measurement covariance matrix R. The Kalman gain was calculated as shown in Equation 3.37 where P^- is the covariance prior to the Kalman update, $\hat{x}^-$ is the estimated state prior to the update, H is as defined in Equation B.17, and G is a two-by-two identity matrix. The measurement covariance matrix R is composed of camera measurement noise sigma values as shown in Equation 3.38.

$$K = P^- H^T \left(\underline{\hat{x}}^-\right) \left[H \left(\underline{\hat{x}}^-\right) P^- H^T \left(\underline{\hat{x}}^-\right) + GRG^T\right]^{-1} \tag{3.37}$$

$$R = \begin{bmatrix} \sigma_{\ell_x}^2 & 0 \\ 0 & \sigma_{\ell_y}^2 \end{bmatrix} \tag{3.38}$$

The measurement noise used for the calculation of the R matrix is shown in Table 3.10. The standard deviation of the noise was set to about 10 pixels. The values in Table 3.10 are calculated for a camera with a 40-degree FOV and images that are 1920 pixels wide. These values were also used in the simulation to add normal randomized noise to the residuals throughout each simulation. Table 3.10 also shows the noise values for the GPS measurement. In the simulations that process GPS and camera measurements, GPS measurements are processed at the beginning of the trajectory when the UAV is at an altitude above 40 meters. After the UAV descends below 40 meters, GPS measurements are no longer processed, and camera measurements are processed when they are available.

ℓ_x	ℓ_y	$z_{gps,x}$	$z_{gps,y}$	$z_{gps,z}$
0.0036361 rad	0.0036361 rad	1.0 m	1.0 m	3.0 m

Table 3.10: Measurement noise 3-σ values

The error state vector $\delta\hat{\underline{x}}$ was updated as shown in Equation 3.39. This updated error state was used to correct the errors accumulated in the state vector using Equation 3.5.

$$\delta\hat{\underline{x}}^{+} = K\left[\underline{\tilde{z}} - \underline{h}\left(\hat{\underline{x}}^{-}\right)\right] \tag{3.39}$$

The covariance was updated as shown in Equation 3.40 where P^{+} is the updated covariance resulting from the Kalman update.

$$P^{+} = \left[I - KH\left(\hat{\underline{x}}^{-}\right)\right]P^{-}\left[I - KH\left(\hat{\underline{x}}^{-}\right)\right]^{T} + KGRG^{T}K^{T} \tag{3.40}$$

The camera measurement is driven by zero-mean, white noise, so the residuals should also be zero-mean and white [69]. Figure 3.23 shows a single simulation run of the camera measurement residual updated using the Kalman update. A visual inspection of the figure shows the residual to be sufficiently zero-mean and white.

As time progresses in the simulation, the estimation errors and their associated covariances should settle into a relatively steady state after a sufficient number of cycles of the Kalman update have been run. This occurs for the position, velocity, attitude, and camera rotation estimation errors. This can be seen in Figures 3.25, 3.26, 3.27, and 3.28. The covariances for the accelerometer and gyroscope biases are not affected much by either the GPS or camera measurements, though it can be seen in Figure 3.29 that the accelerometer bias covariance does start to clamp down a bit more as fiducial measurements begin to be processed. The gyroscope bias in Figure 3.30 is extremely small to begin with, and it stays pretty level throughout the simulation.

In all of the plots, the covariance lines respond appropriately to GPS and fiducial measurements, and the Monte Carlo simulations stay sufficiently within the 3-σ bounds.

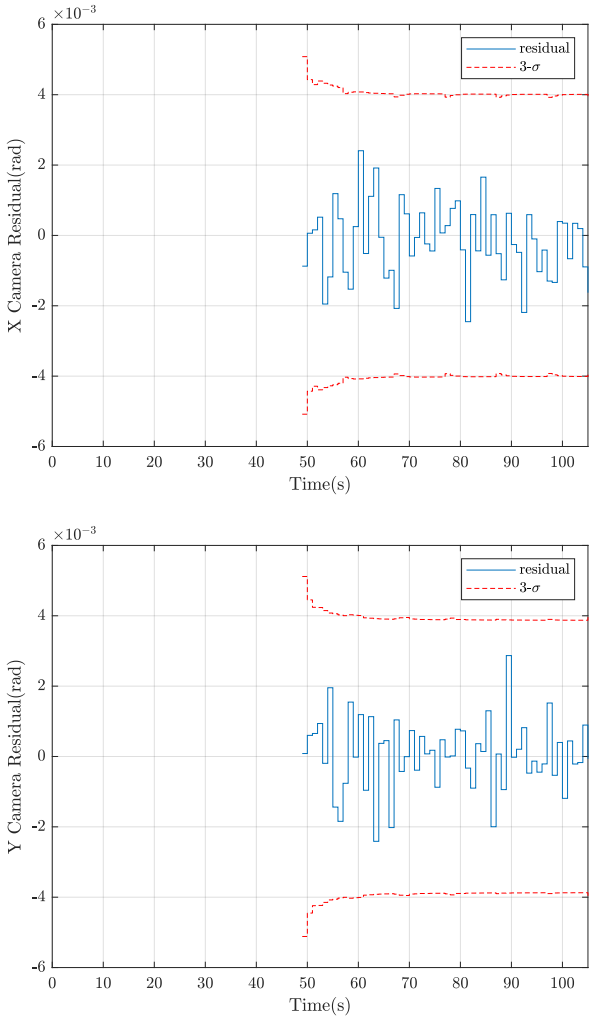

Fig. 3.23: X and Y camera measurement residuals propagated along with the 3-σ values

This validates that the covariance is updating properly with the added discrete GPS and camera measurements. The G and R matrices have also been validated.

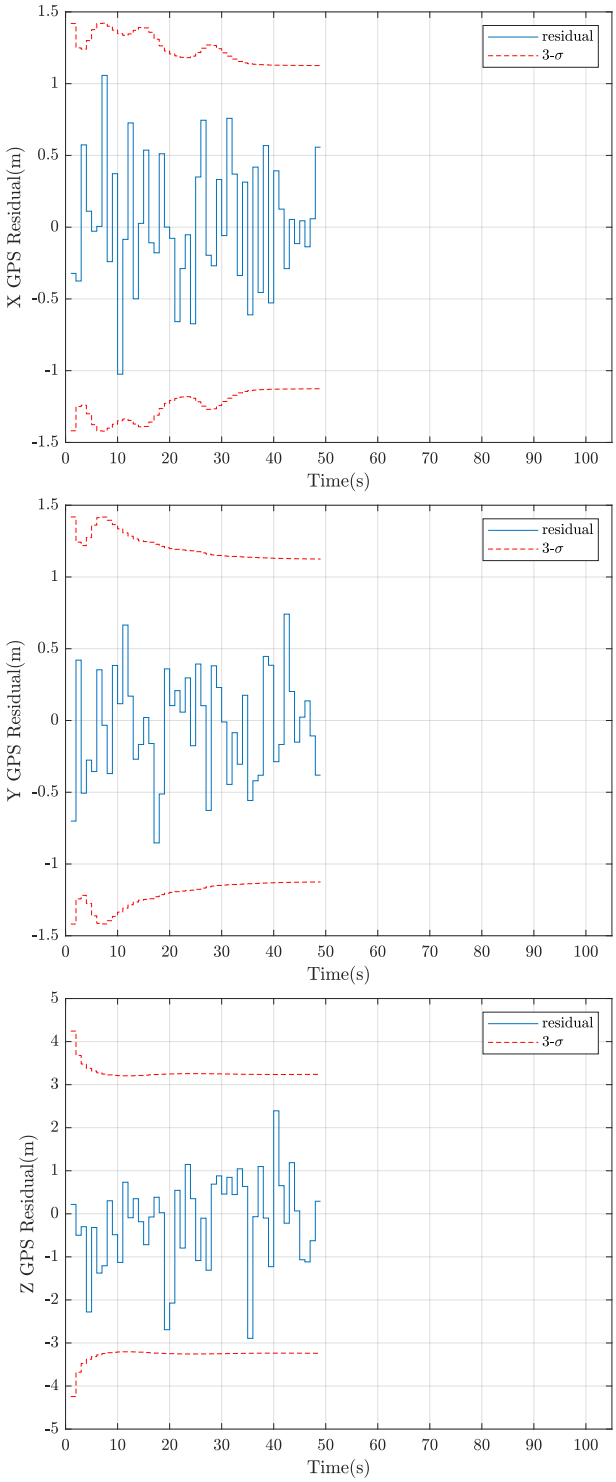

Fig. 3.24: X, Y, and Z GPS measurement residuals propagated along with the 3-σ values

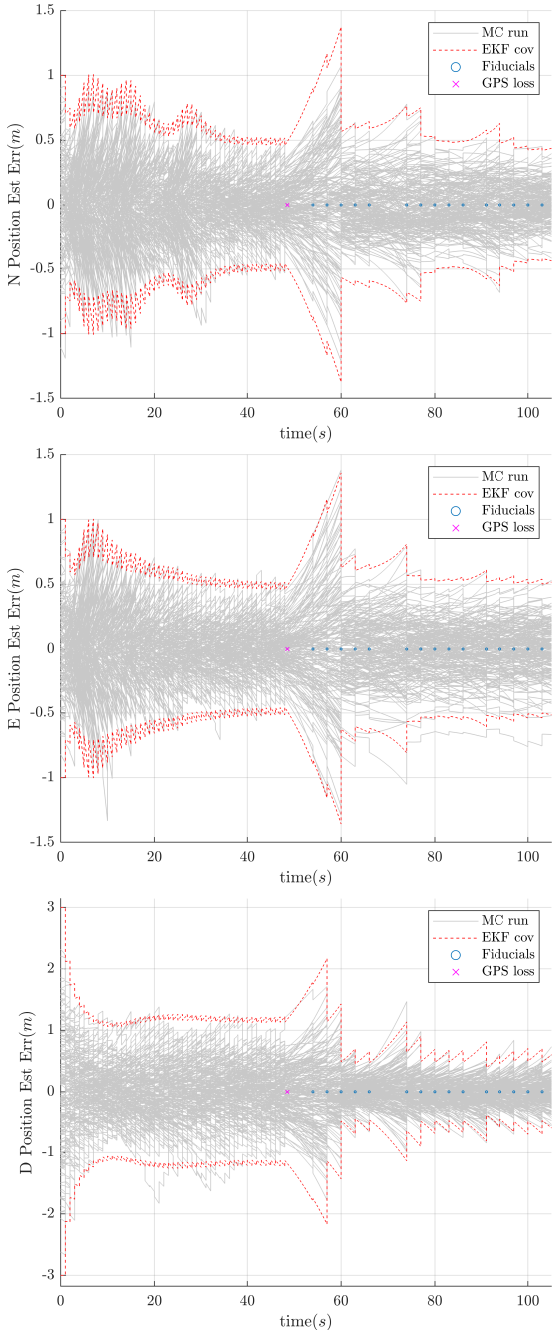

Fig. 3.25: North, east, and down position estimation error with Kalman update

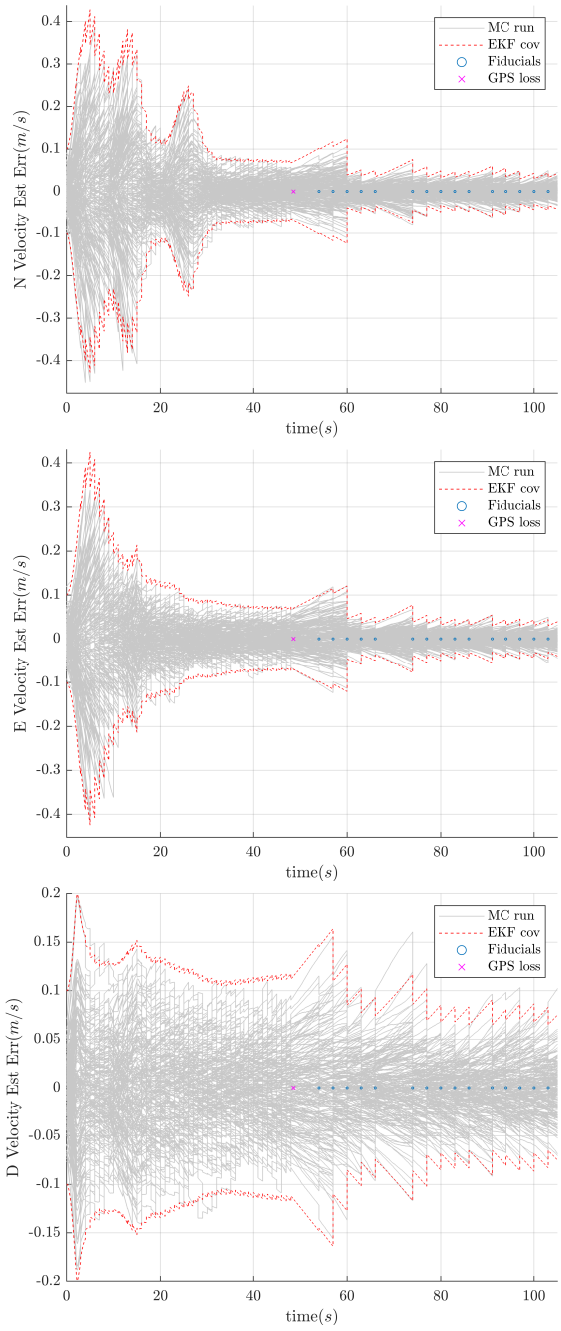

Fig. 3.26: North, east, and down velocity estimation error with Kalman update

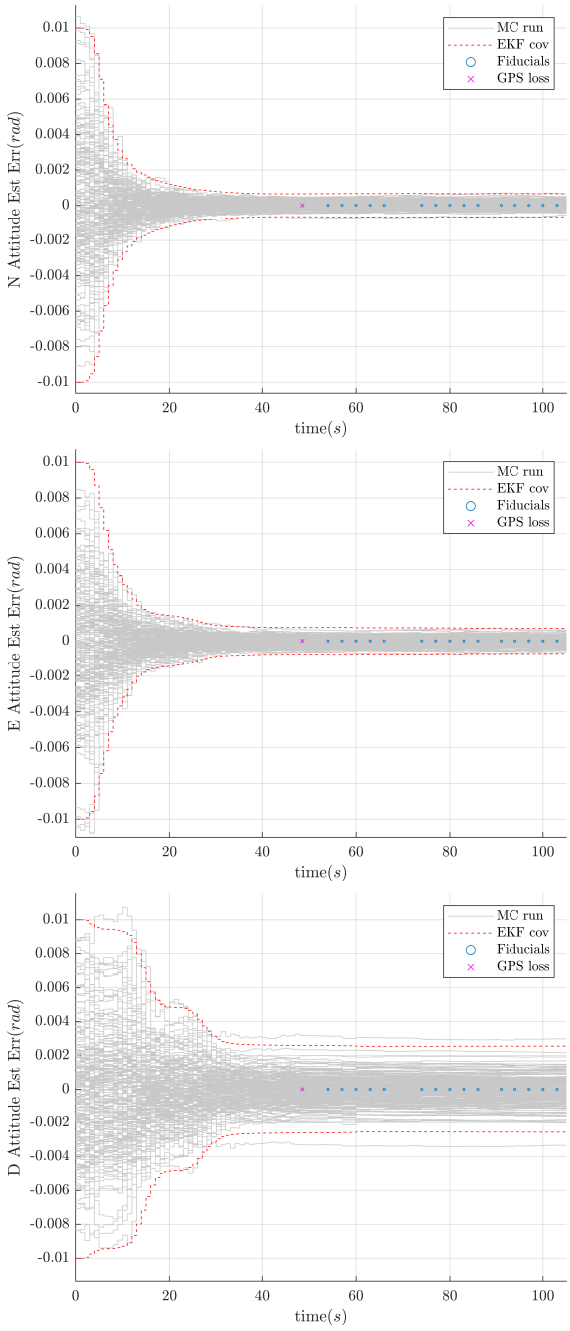

Fig. 3.27: North, east, and down attitude estimation error with Kalman update

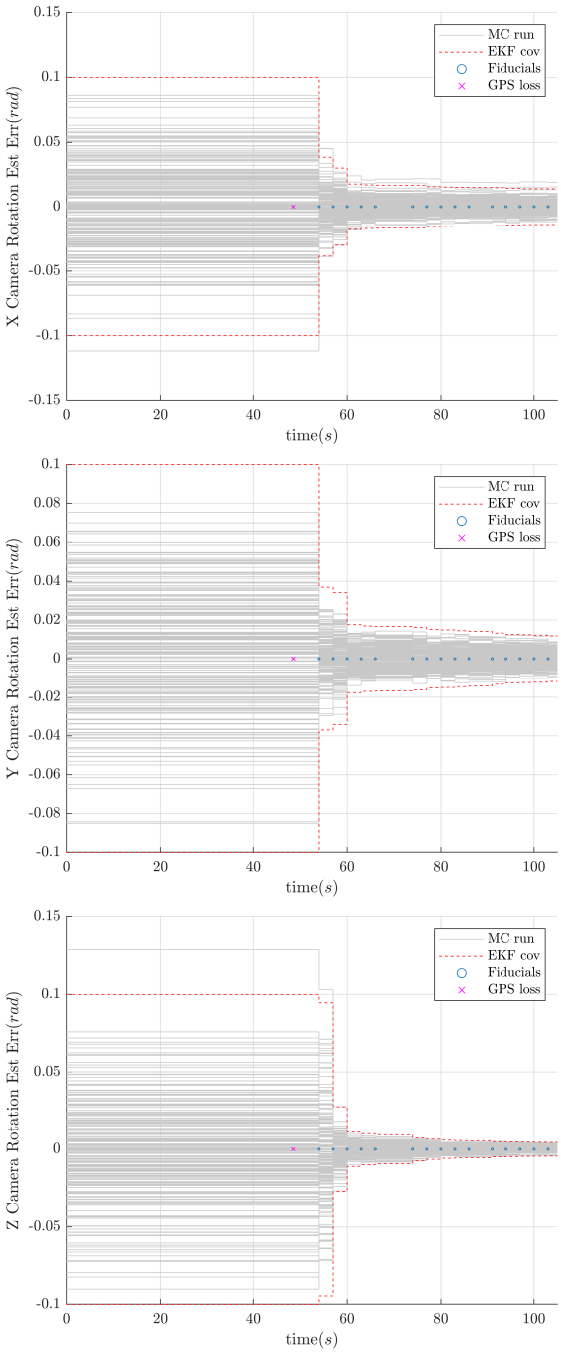

Fig. 3.28: X, Y, and Z camera attitude estimation error with Kalman update

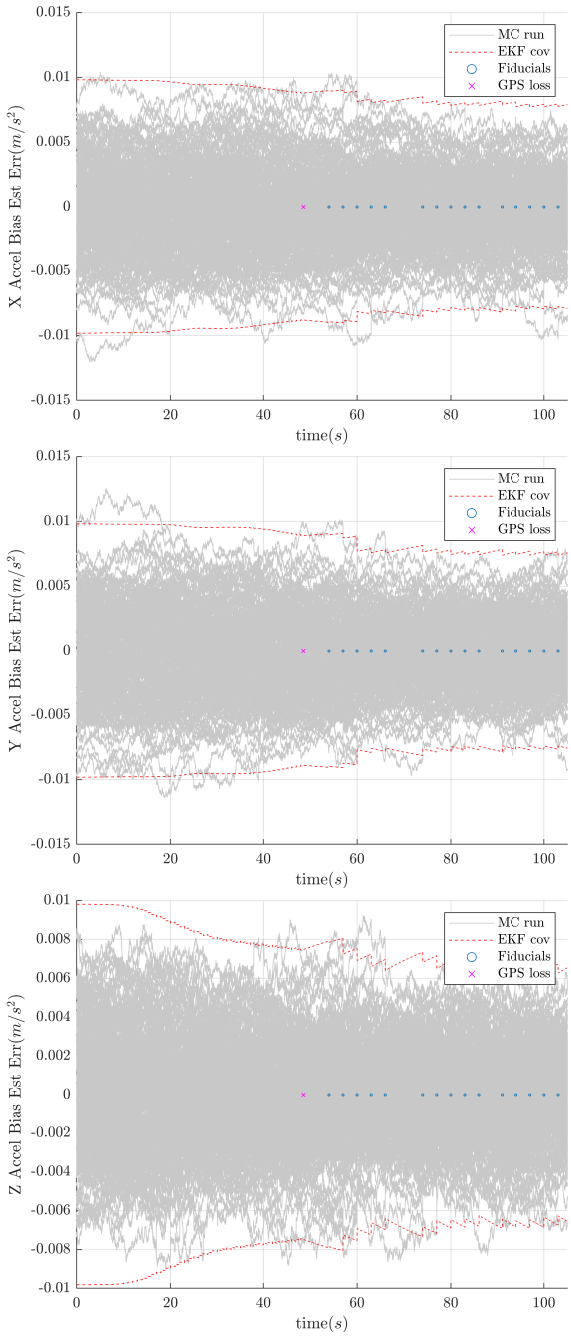

Fig. 3.29: X, Y, and Z accelerometer bias estimation error with Kalman update

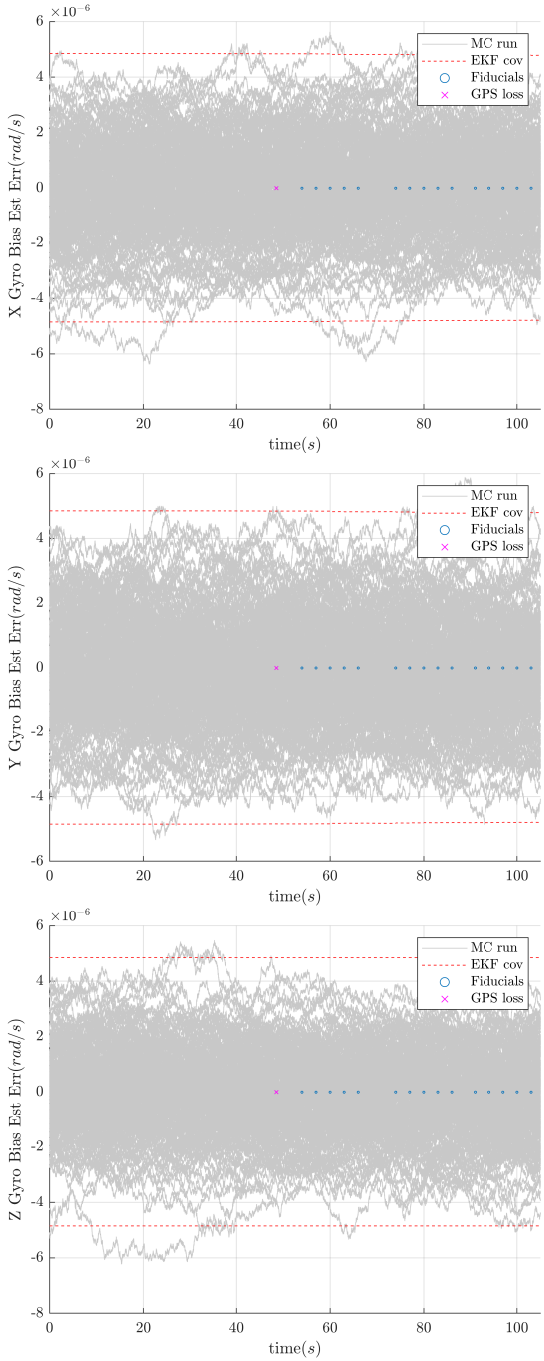

Fig. 3.30: X, Y, and Z gyro bias estimation error with Kalman update

CHAPTER 4

SIMULATION ANALYSIS

In this chapter, the effect of variables on the EKF's estimation capability is presented and analyzed. Commercial-, tactical-, and navigation-grade IMUs were simulated by varying associated parameters such as steady-state standard deviation, time constants, and random walk values to match those of typical commercial-, tactical-, and navigation-grade IMUs. Fiducials were simulated as individual points in a camera's image. The simulated EKF was propagated with a simulated flight trajectory using the varying IMU parameters, varying placements of fiducials along the trajectory path, various distances at which the fiducials lay below the UAV's trajectory, and various image sampling frequencies. The sensitivity of the EKF to the various configurations was analyzed by observing the resulting estimation error variance.

Fiducials were placed a specified distance below the UAV's trajectory and on alternating sides of the trajectory path 6 meters off the side of the path. This was done to simulate having the fiducials on the top of light posts, traffic signals, or the side of buildings on either side of an urban canyon. The trajectory ran for 105 seconds and is shown in Figure 4.1. This trajectory starts with some S-curves to exercise the IMU sensors and the UAV flight dynamics. Above 40 meters, GPS measurements were processed. Once the UAV descended below 40 meters at approximately 48.5 seconds, GPS measurements were stopped, and only LOS measurements were active.

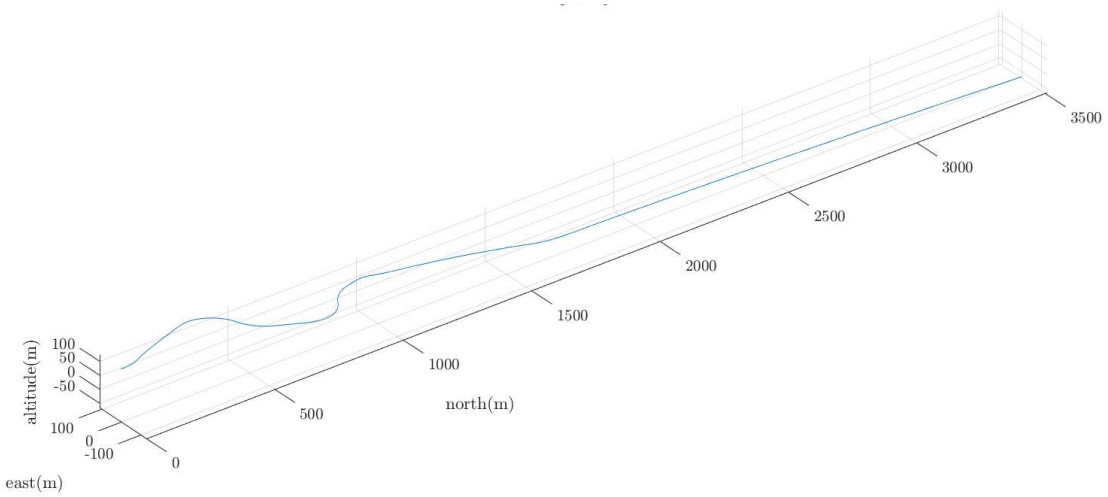

Fig. 4.1: Trajectory used to analyze fiducial density and IMU grades

The camera model used has a FOV of 40 degrees. Therefore, FOV testing was done to see which fiducial(s) were within the camera's frame during each Kalman cycle update. If no fiducials were within view, no LOS measurement was processed. If fiducial(s) were in view, measurements to each fiducial were processed as separate measurements. The camera frame was set to be aligned with the UAV's body frame. This setup is shown in Figure 4.2 [9], [10].

IMU Grade	Commercial	Tactical	Navigation
$\sigma_{ss,a}(g)$	0.0100	0.0010	0.0001
$VRW(m/s/\sqrt{hr})$	0.600	0.060	0.006
$\sigma_{ss,g}(deg/hr)$	10.0	1.0	0.1
$ARW(deg/\sqrt{hr})$	0.700	0.070	0.007

Table 4.1: IMU specifications by grade

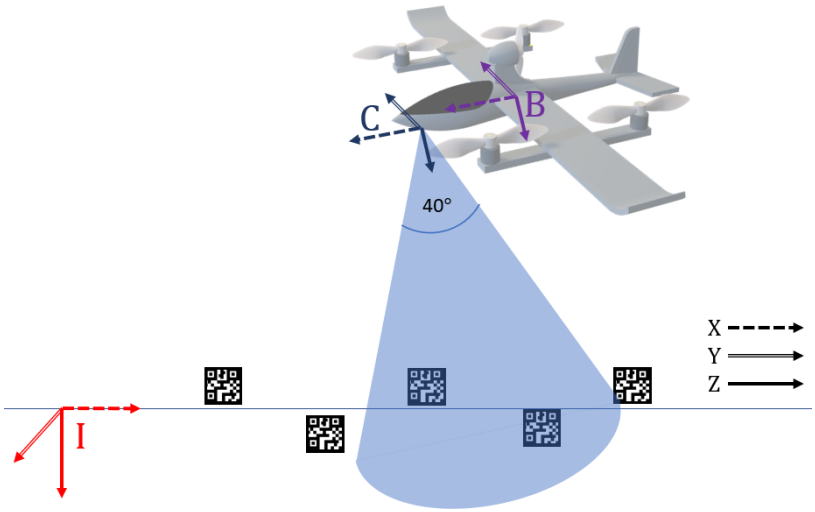

Fig. 4.2: Field of view testing

A simulation of the EKF processing measurements and estimating states was run for fiducial densities starting at 10 meters and going up to 400 meters by 10-meter increments. This was done for three grades of IMU: commercial, tactical, and navigation. The specifications for these three grades of IMU are shown in Table 4.1. Time constants τ_a and τ_g were set to 60 seconds each for all of the simulations. The image frequency was set to 5Hz, and the distance to the fiducials was set to 15 meters for the fiducial spacing and IMU grade analysis.

Figures 4.3 and 4.4 show the resulting plots from some of these simulations for the north position estimation error. The plots show 200 grey hairlines that correspond with the estimated north position error for 200 Monte Carlo runs of the simulation along with the

red 3-σ lines. They also show a red X on the x-axis to mark the time at which the UAV descended into the urban canyon, and GPS measurements were lost or deemed unreliable. Blue dots along the x-axis mark times during the simulation at which a LOS meausrement to a fiducial was processed. For brevity and space, only these samples of the north position estimation error are shown. Because the UAV is mostly moving in the north direction, this was the direction chosen to best represent the results.

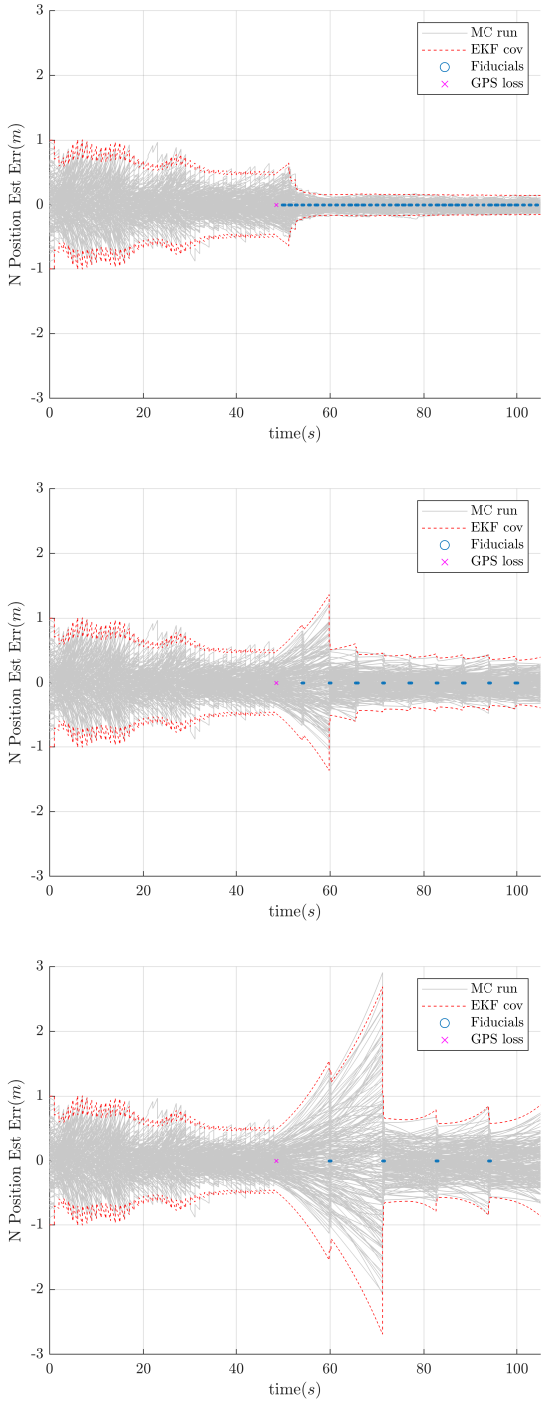

Fig. 4.3: Estimation error plots for 50- (top), 200- (middle), and 400-meter (bottom) fiducial densities using tactical-grade IMU

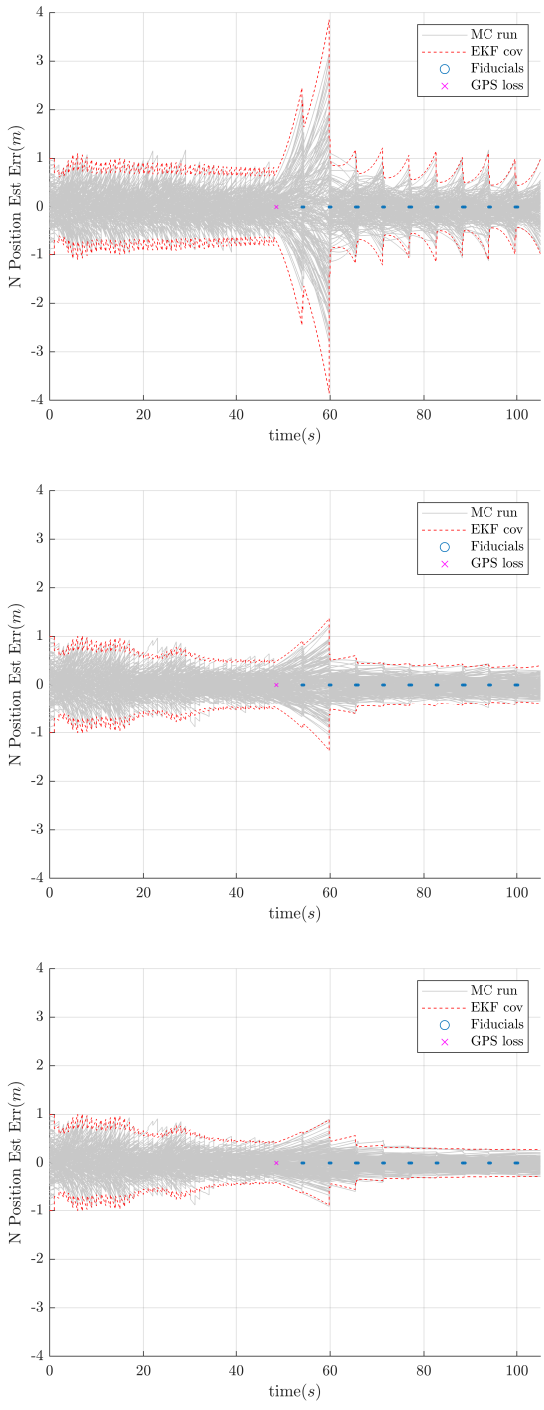

Fig. 4.4: Estimation error plots for 200-meter fiducial density using commercial- (top), tactical- (middle), and navigation-grade (bottom) IMUs

Several interesting observations came from viewing the estimation error plots in Figures 4.3 and 4.4. In Figure 4.3, it can be seen that the point where GPS measurements were no longer processed can be seen in the plots where the 3-σ bounds start to grow before reaching the fiducials. Another observation is that the 3-σ lines clamped down after each fiducial was detected. For the 50-meter fiducial density, the error does not grow significantly between fiducial measurements, and the estimated error is fairly constant. The 200-meter spacing allows for estimation error to grow between fiducial sightings, though by less than 0.1 meters. The estimated error with 400-meter spacing grows more than with the 200-meter spacing, but not without bounds.

In Figure 4.4, it can be seen that the estimation error with the commercial-grade IMU grows to just outside the bounds which are maintainable by the GPS measurements. The tactical-grade IMU keeps the error within approximately the same range as the GPS measurements. The navigation-grade IMU allows for estimation error even tighter than that of the tactical-grade IMU, and the estimation errors are even less than those obtained when only GPS updates were processed.

The average 3-σ value (the red lines in the estimation error plots) was calculated for each simulation for the last third of the simulation (from 69.995 seconds to 105 seconds). This portion of the simulation was chosen because the 3-σ values settled to a consistent pattern by this time and would represent the estimation error obtainable with the given configuration well. Figure 4.5 shows these average 3-σ errors as the space between fiducials was varied and as the grade of IMU was varied. During these simulations, the fiducials were placed 15 meters below the UAV's trajectory, and images were processed at 5 Hz.

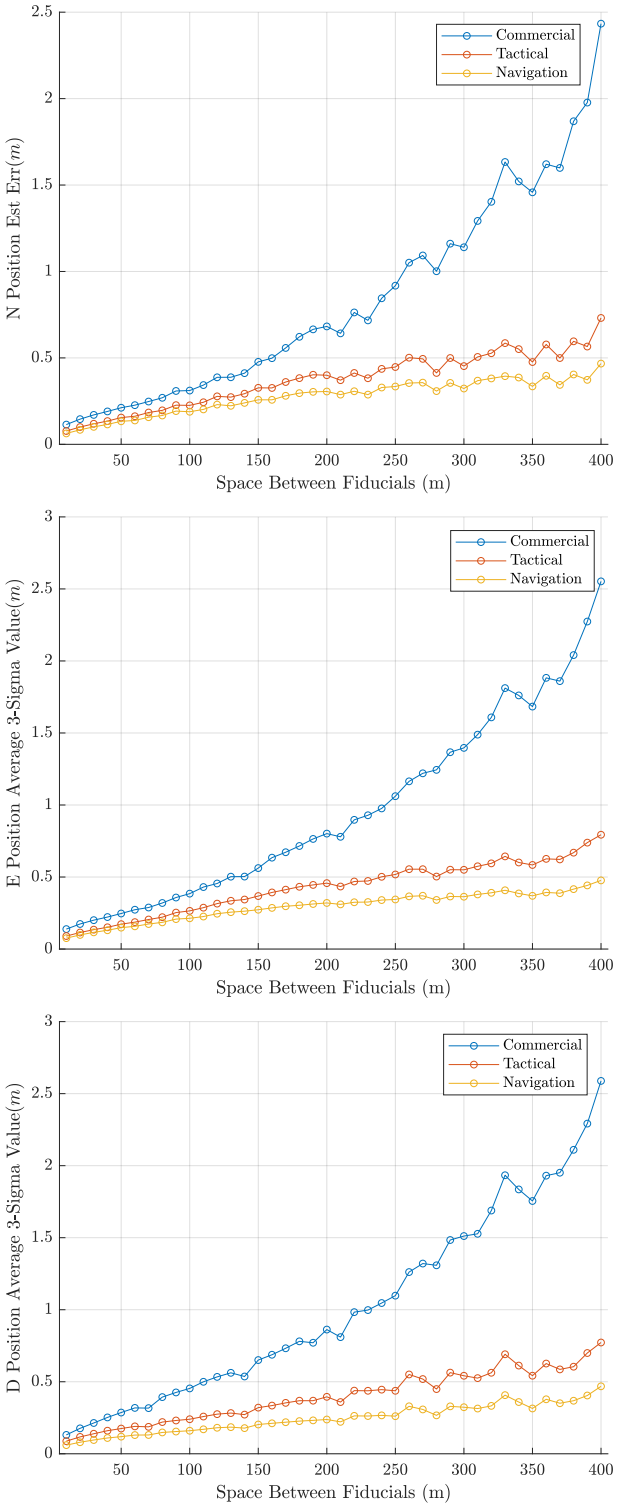

Fig. 4.5: Average 3-σ steady state position estimation error for three grades of IMU and varied distance between fiducials for the three axes of the NED frame

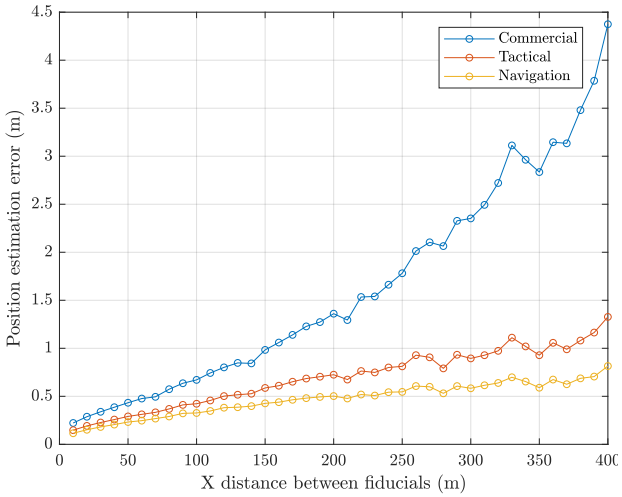

Fig. 4.6: Combined average 3-σ steady state position estimation error for three grades of IMU and varied distance between fiducials

Figure 4.6 shows the result of combining all three directions of position estimation error plots into one summary plot. These values were obtained by taking the square root of the sum of the squares of the three position directions.

It can be seen in Figures 4.5 and 4.6 that the position estimation errors increase exponentially as the fiducial density decreases for the commercial-grade IMU. This may also be the case for the tactical and navigation-grade IMUs, but for the range of parameters tested, these relationships are more linear. The average estimation error is not drastically different between the three grades of IMU at the tighter densities of fiducials. However, when the fiducials are spaced further apart, the commercial-grade IMU does not allow as tight of state estimations as those of the tactical- and navigation-grade IMUs. The tactical-grade IMU also does not allow for as tight of state estimations as the navigation-grade IMU, however, the difference between the tactical- and navigation-grade IMUs is much smaller than the difference between the commercial- and tactical-grade IMUs. The difference in the estimation error using the tactical- and navigation-grade IMUs is up to about 0.5 meters at

the 400-meter spacing range. The difference between estimation errors using commercial- and tactical-grade IMUs, on the other hand, is up to 3 meters.

From this analysis, if less than half a meter of error is required, it is recommended that fiducials be placed no more than 50 meters apart if a commercial-grade IMU is used or no more than 120 meters apart if a tactical-grade IMU is used. From this analysis, a navigation-grade IMU does not give a lot of benefit over a tactical-grade IMU, and it is not recommended unless necessary because the cost of navigation-grade IMUs is significantly higher than that of tactical-grade IMUs. In many cases, it would be more cost-effective to have more densely-spaced fiducials.

Figures 4.7 and 4.8 show the average position estimation 3-σ values as the vertical distance above fiducials was varied between 5 meters and 50 meters. During these simulations, the IMU was set to use tactical-grade specifications, fiducials were spaced 100 meters apart, and the image capture frequency was set to 5 Hz. While it can be seen that the estimation error grows as the distance to the fiducials grows, it can also be seen that there is a jump between about 16 meters and 18 meters. To investigate the cause of this jump, the estimation error plots were generated for these two distances. These plots can be seen in Figure 4.9. While it it can be seen that fiducials are skipped over in both scenarios, the 18-meter distance allows for an additional fiducial to be processed, and this made all the difference. This demonstrates the importance of considering trajectories and flight dynamics when determining a good fiducial placement strategy. A fiducial does no good if the UAV is moving so quickly that the image processing rate causes the fiducial to be missed or the UAV is at an attitude such that the camera is not pointed toward the fiducial. This also demonstrates how quickly the estimation error can grow if even a single fiducial measurement is missed. To reduce the possibility of estimation errors growing beyond an acceptable amount, it could be good to place fiducials twice as densely as is calculated to be necessary in case a fiducial is missed here and there.

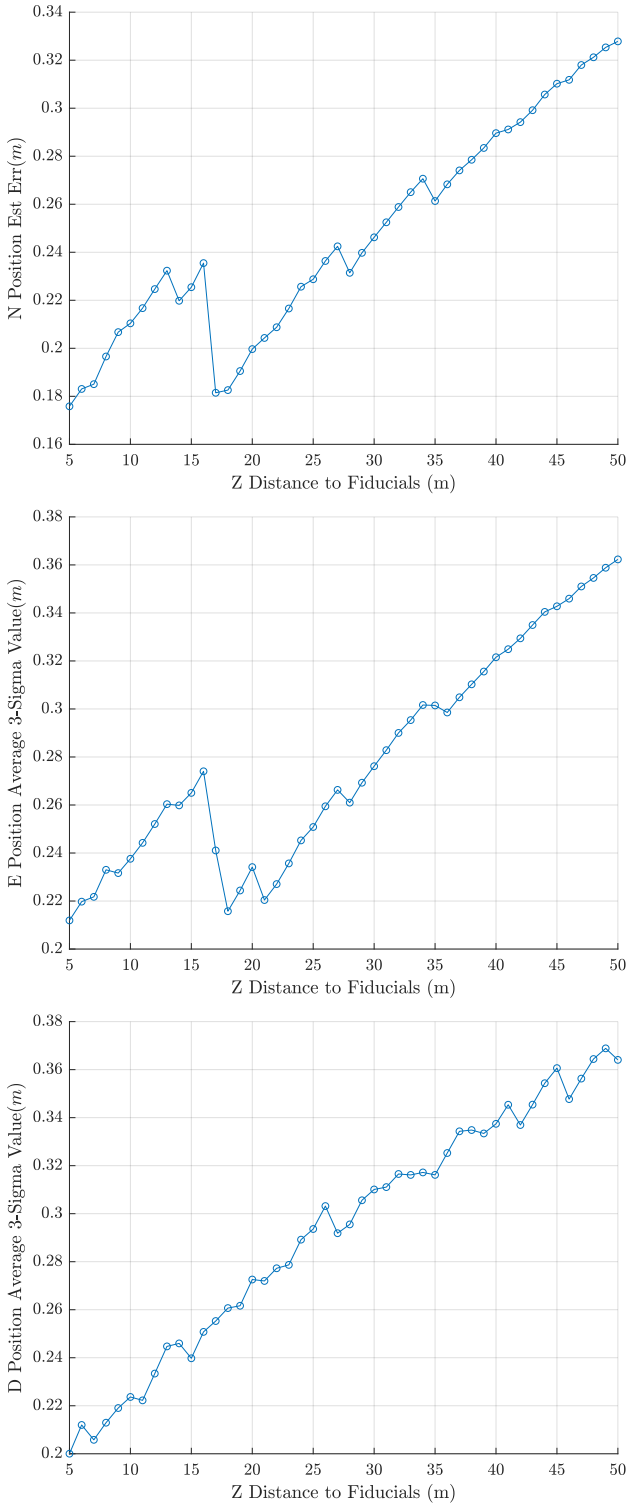

Fig. 4.7: Average 3-σ steady state position estimation error for the three axes of the NED frame as height of the trajectory above the fiducials was varied

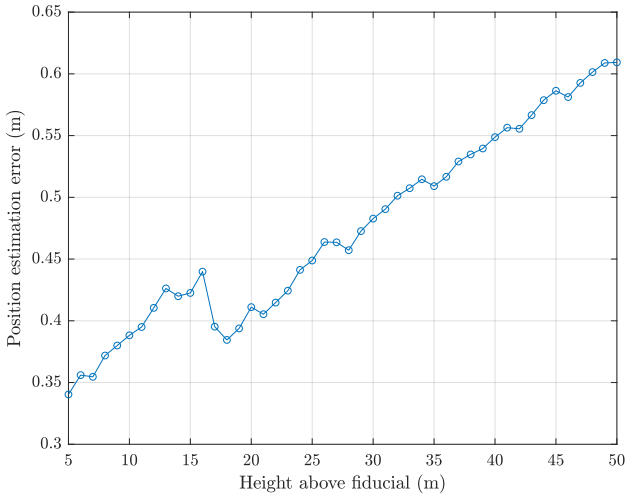

Fig. 4.8: Combined average 3-σ steady state position estimation error as height of the trajectory above the fiducials was varied

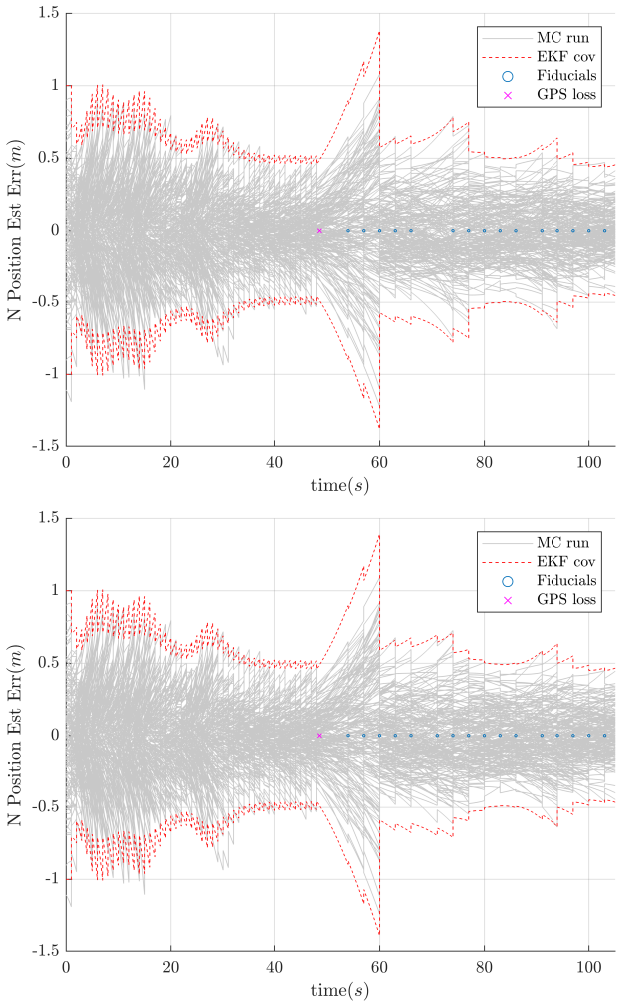

Fig. 4.9: Estimation error plots for north position estimation with fiducials at 16 m (top) and 18 m (bottom)

Figures 4.10 and 4.11 show the effect of varying the frequency at which images are captured and processed. During these simulations, a tactical-grade IMU was used, fiducials were spaced 100 meters apart, and fiducials were placed 15 meters below the UAV's trajectory. At higher frequencies (lower times between image samples), a somewhat consistent relationship between position estimation error growth and frequency variance appears to be forming. However, as the frequency decreases toward 1 Hz, this relationship begins to be more sporadic. When the image capture frequency is higher, it is more likely that every

fiducial will be processed, possibly multiple times each. However, at the lower frequencies, it is more likely that fiducials will be missed altogether. At that point, it is simply chance as to whether the camera was capturing an image at just the right time to process a fiducial measurement. This indicates that as the image processing frequency increases, the system becomes less sensitive to fiducial placement.

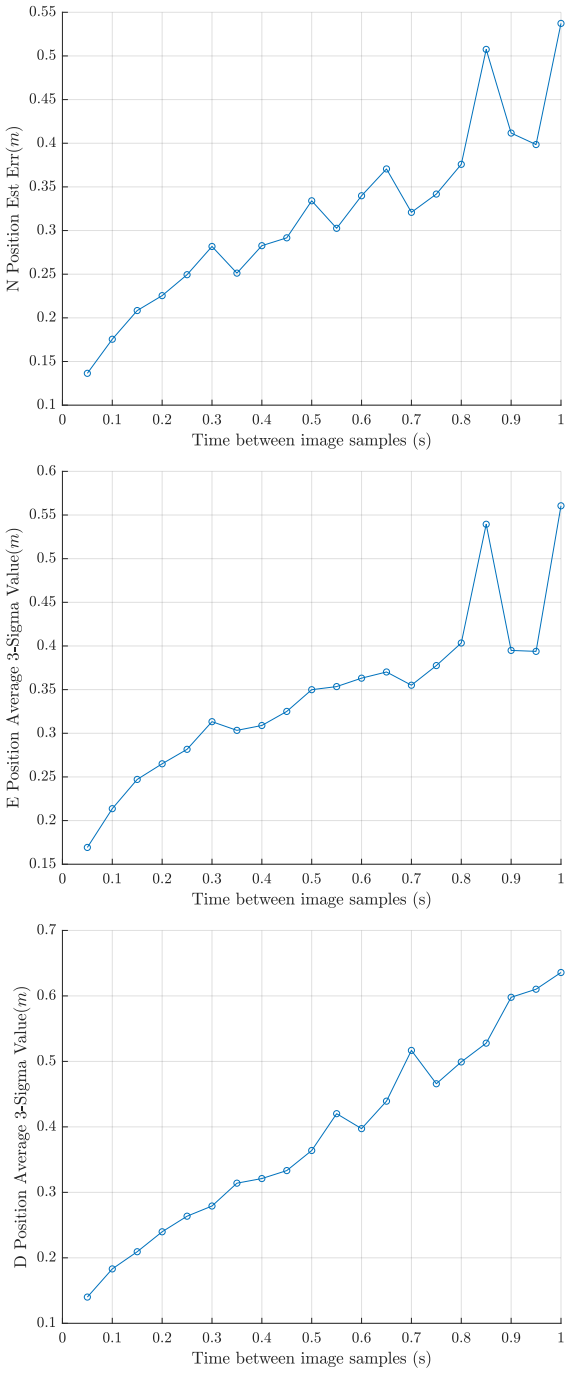

Fig. 4.10: Average 3-σ steady state position estimation error for the three axes of the NED frame as frequency of image processing was varied

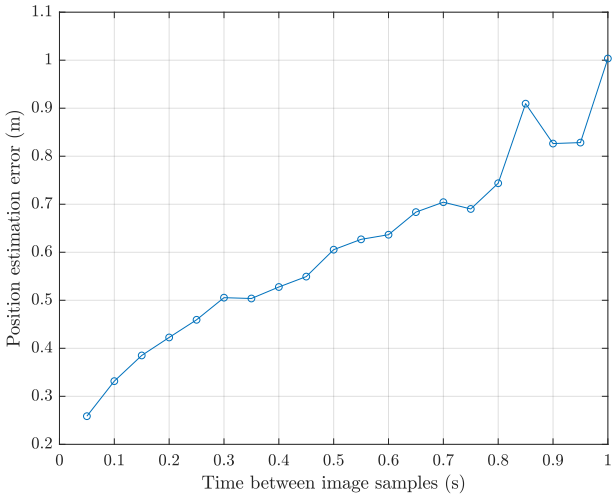

Fig. 4.11: Combined average 3-σ steady state position estimation error as frequency of image processing was varied

CHAPTER 5

HARDWARE DEMONSTRATION PREPARATION

After the EKF was developed and tested through simulation, preparations were made for implementing the system in hardware. Various fiducial types were surveyed, and ArUco markers were recommended for the purpose of this study. A method for extracting a single-pixel point from the ArUco markers was developed. Hardware implementation plans include using an OptiTrack™ motion capture system with Motive™ software to obtain truth data for the position and attitude of the UAV. As part of this study, the noise associated with the motion capture system was characterized, and a method for reducing the magnitude of the noise was designed, implemented, and tested.

5.1 Fiducial Survey and Selection

In Faessler et al. [63] and Censi et al. [62], LEDs flashing at specific frequencies were used as fiducials. In Censi et al. [62], it was mentioned that the OptiTrack system that was used interfered with the LED fiducials because the OptiTrack system itself uses flashing lights to reflect off of markers on rigid bodies to track them. Because an OptiTrack system is integral to obtaining truth data for the research in this paper, using LEDs as the fiducials was ruled out as a possibility. However, the research in this paper could be applied to LED fiducials because the fiducials are reduced to a single point before being processed as measurements for the Extended Kalman Filter.

In Cesar et al. [65], ARToolKit, AprilTag, and ArUco marker fiducials were compared in an underwater environment. That research concluded that AprilTags performed the best in underwater environments as far as clarity and accuracy of data goes, but AprilTags took the longest to process with almost double the average processing time as that required for ArUco markers which were the fastest for processing time. The ArUco markers were comparable to the AprilTag markers in other areas, particularly in clear-water conditions,

which would be most similar to how they would be used in this research, with air as the medium between the camera and the fiducial.

In this research, three different fiducials were compared by using corresponding Robot Operating System (ROS™) packages to process them using a laptop with an Intel® CORE™ i7 processor. Images of the fiducials were captured using a FLIR Firefly S® camera. The three fiducial types were AprilTags [70], QR Codes [71], and ArUco markers [72]. AprilTags were able to be published at a rate of 7 Hz, QR codes at a rate of 3 Hz, and ArUco markers at a rate of 19 Hz. It was also observed that the detection of the QR codes did not work if the QR code was tilted within the image more than a few degrees.

AprilTags can contain up to 12 bits of data [47], supporting 4,095 different values. Various types of ArUco marker typically support up to 1000 different values, which is about 10 bits of data. While the amount of data that can be represented in ArUco markers is less than that of AprilTags, the quicker processing time associated with the ArUco markers was considered to be the most important factor for this study. As demonstrated in Chapter 4, the frequency at which images can be processed drastically affects how much the fiducials aid state estimation. Therefore, ArUco markers were chosen as the fiducials for this research.

Compressed raw images of ArUco markers were gathered using the Flir Firefly camera at various distances and rotations using ROS. These images were processed using the ArUco detection software built on ROS [72]. This detection produced pixel coordinates of the four corners of the ArUco markers. For each detection, the four corners were used to construct two intersecting lines. The point of intersection between the two lines can be used as the single-pixel location for the fiducial camera measurements. The points reported as the four corners of the ArUco markers were always in sequential order in a clockwise direction as shown in Figure 5.1. Therefore, the equations for the two lines in the form $y = mx + b$ can be seen in equations 5.1 to 5.4. The intercept pixel point (x,y) was then found according to equations 5.5 and 5.6.

$$m_1 = \frac{Y_0 - Y_2}{X_0 - X_2} \tag{5.1}$$

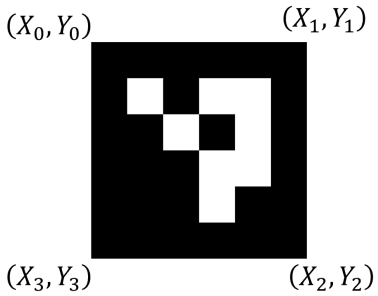

Fig. 5.1: ArUco marker corner sequence reported by ROS ArUco detection software

$$b_1 = Y_0 - m_1 * X_0 \tag{5.2}$$

$$m_2 = \frac{Y_1 - Y_3}{X_1 - X_3} \tag{5.3}$$

$$b_2 = Y_1 - m_1 * X_1 \tag{5.4}$$

$$x = \frac{b_1 - b_2}{m_2 - m_1} \tag{5.5}$$

$$y = m_1 * x + b_1 \tag{5.6}$$

After the four corners and the corresponding calculated center of the ArUco marker were determined, corresponding pixels in the original images were set to white. The corners and center were reported with decimal precision. For the purpose of displaying the values on the images, these decimal values were rounded up to the next pixel value. Figure 5.2 shows one of the processed images with the corner and center pixels changed to white beside a zoomed-in version of the image to better show the pixel determined as the center pixel.

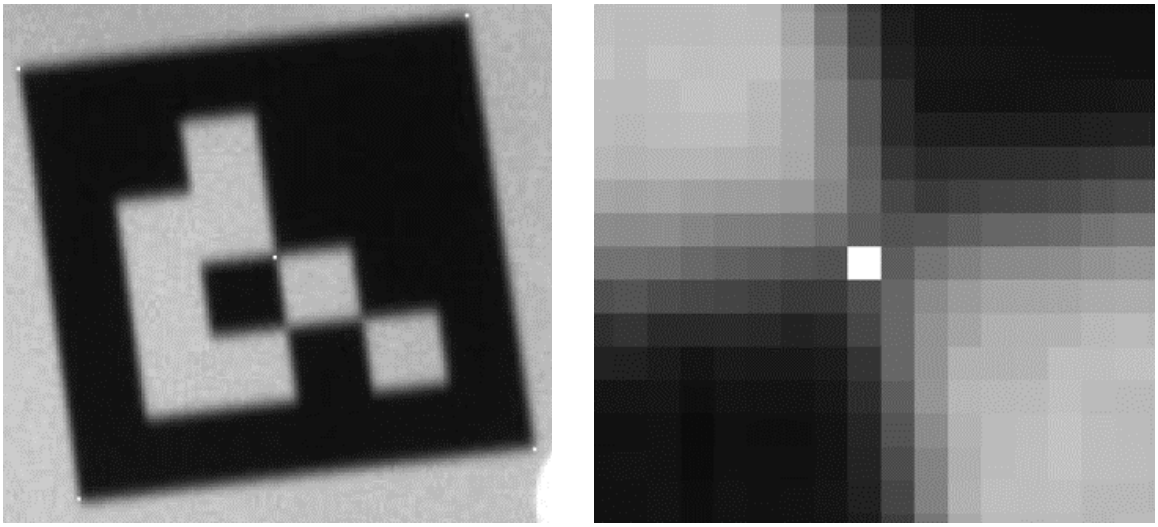

Fig. 5.2: Example of ArUco marker image with corners and center marked next to the zoomed-in center of the same image

Figure 5.3 shows additional examples of ArUco markers with the corners and center marked. These figures show that the center pixel of the ArUco marker can be determined accurately using Equations 5.1 to 5.6 with the four corners as reported from the ROS ArUco detection software.

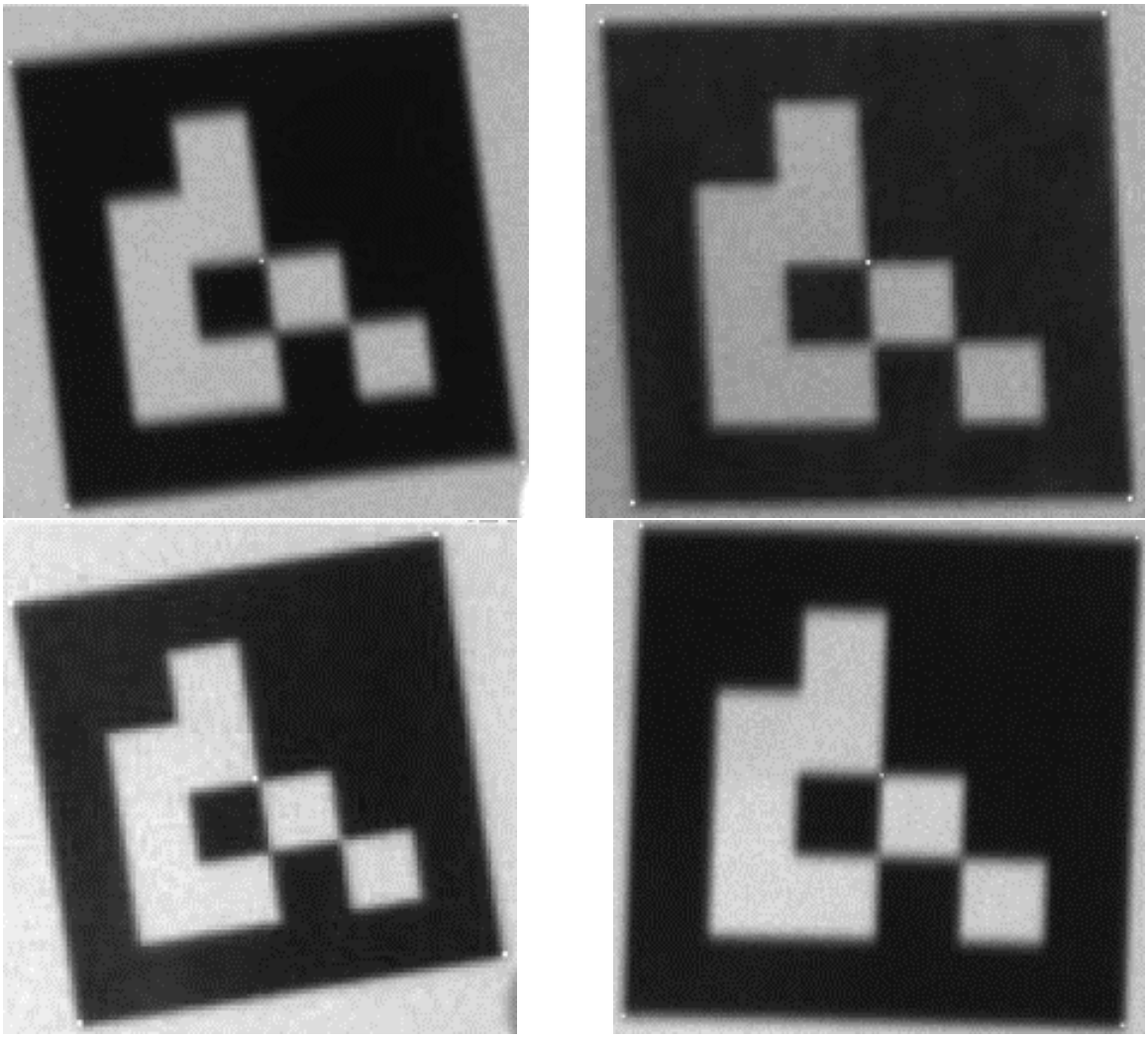

Fig. 5.3: Additional examples of corner detection and center calculation for ArUco markers

5.2 Motion Capture Noise Characterization

To test the effectiveness of the EKF developed in this research, a motion capture system can be used to gather the position and attitude of the UAV. These measurements can be used as the truth position and attitude, which can then be compared to the position and attitude states estimated by the EKF as it propagates its states using IMU measurements and updates its states using LOS measurements to fiducials. Noise can also be added to the motion capture data to be used as simulated GPS signals.

To test the noise associated with the motion capture system, reflective markers were

placed on an S500 quadcopter frame. This frame was then set on the floor approximately in the center of the motion capture area, and the position and attitude of the UAV were collected over time. The quadcopter was not moved during the test, and nothing else was moving within the motion capture area throughout the duration of the test. In the original configuration, reflectors were contained within a radius of about 170 mm away from the center of the UAV. In an attempt to reduce the amount of noise associated with the motion capture position and attitude measurements, new legs and attached adjustable extenders were designed (shown in Figure 5.4).

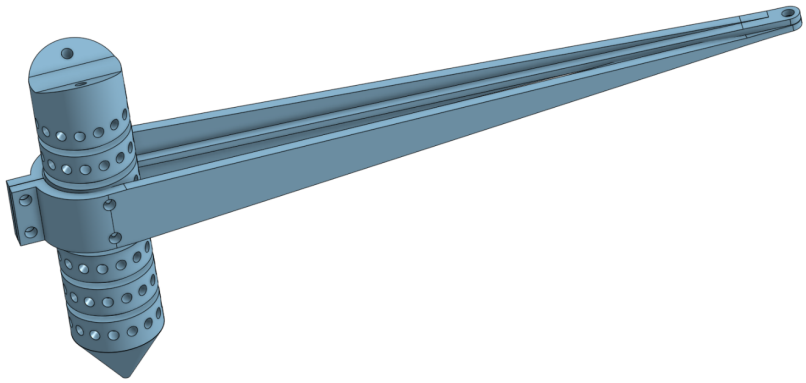

Fig. 5.4: UAV support leg with reflective marker extender beams

These extenders were attached to the quadcopter frame so that the distance between the reflective markers was increased. The extended beams made the total maximum radius of the ArUco marker placement about 270 mm in the horizontal plane. The maximum range in the vertical plane was about 80 mm from the highest reflective marker to the lowest for both configurations. Figure 5.5 shows the configuration of the reflectors before the extenders were added, and Figure 5.6 shows the configuration of the reflectors after the extenders were added.

For each configuration, the position of the UAV was plotted in the x, y, and z directions. The change in the attitude from the first reported attitude was also plotted. The attitude change from the original attitude to the attitude at time i was calculated as shown in

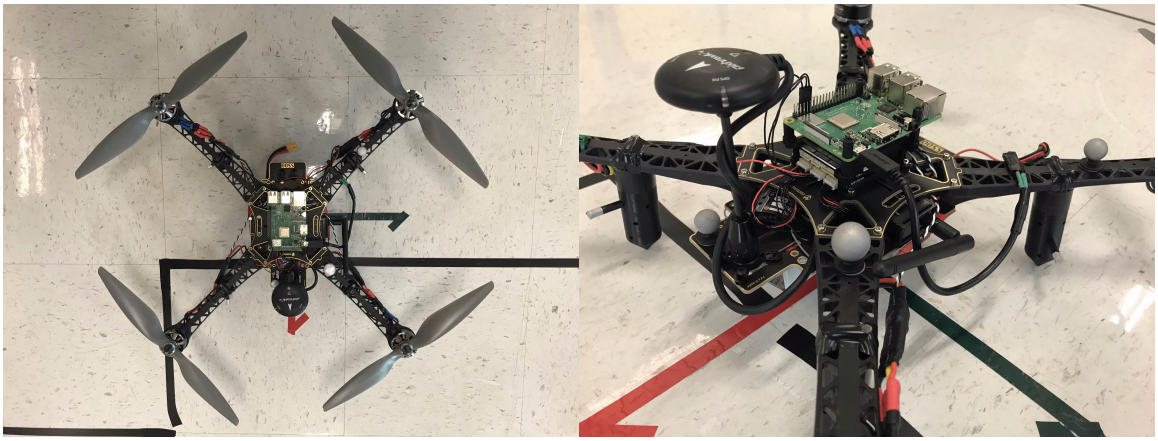

Fig. 5.5: Configuration for motion capture reflective markers without extenders

Fig. 5.6: Extended configuration for motion capture reflective markers

equations 5.7 and 5.8. The results are shown in Figures 5.7 to 5.12 side-by-side. Note that the baseline test for the narrower reflective marker placement was only run for about 67 seconds while the test with the broader reflective marker placement was run for about 490 seconds. The position and attitude plots were adjusted by taking the center position and attitude change in each plot and subtracting the center values from each value before plotting them so that the axes would be the same for both configurations' plots for ease of comparison.

$$\delta q_b^n = q_b^n\left(t_1\right) \otimes q_b^n\left(t_i\right)^* \tag{5.7}$$

$$\delta\theta = 2\delta q_b^n\left(2:4\right) \tag{5.8}$$

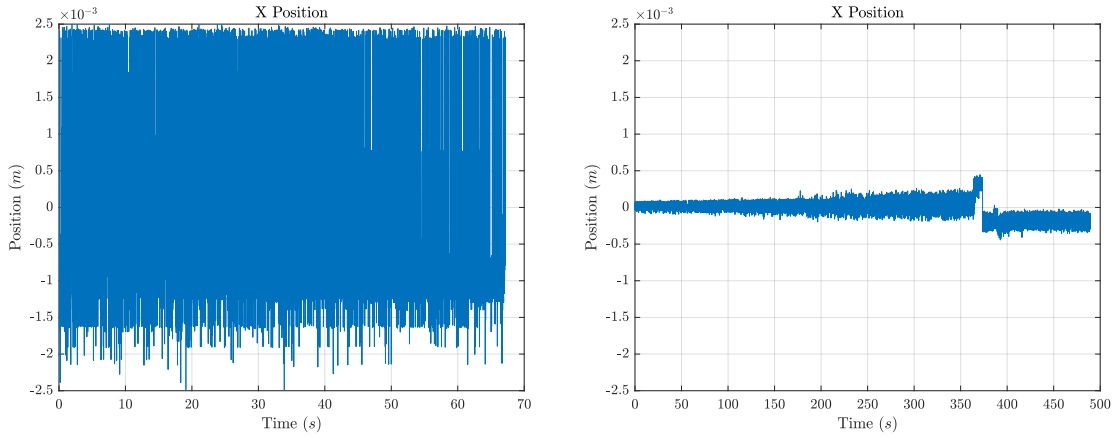

Fig. 5.7: Position in the navigation X direction during baseline test for narrow reflector placement (left) and wide reflector placement (right)

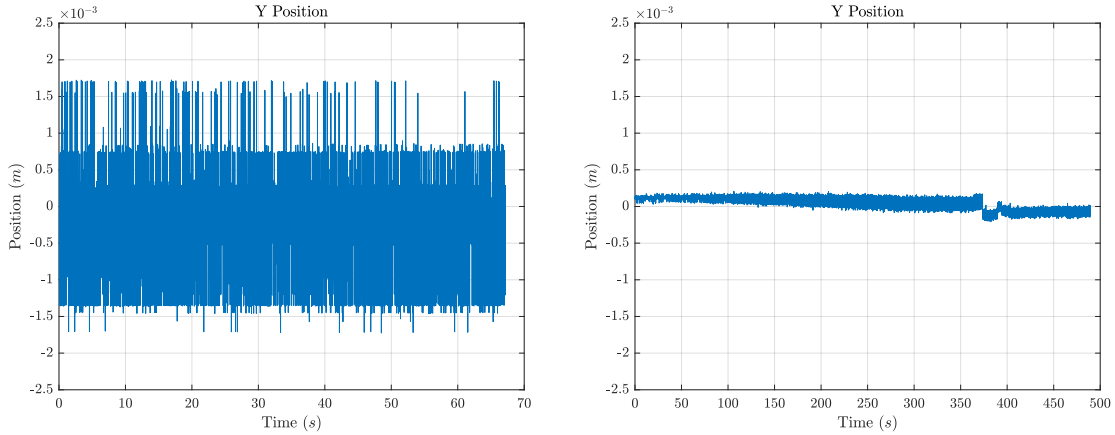

Fig. 5.8: Position in the navigation Y direction during baseline test for narrow reflector placement (left) and wide reflector placement (right)

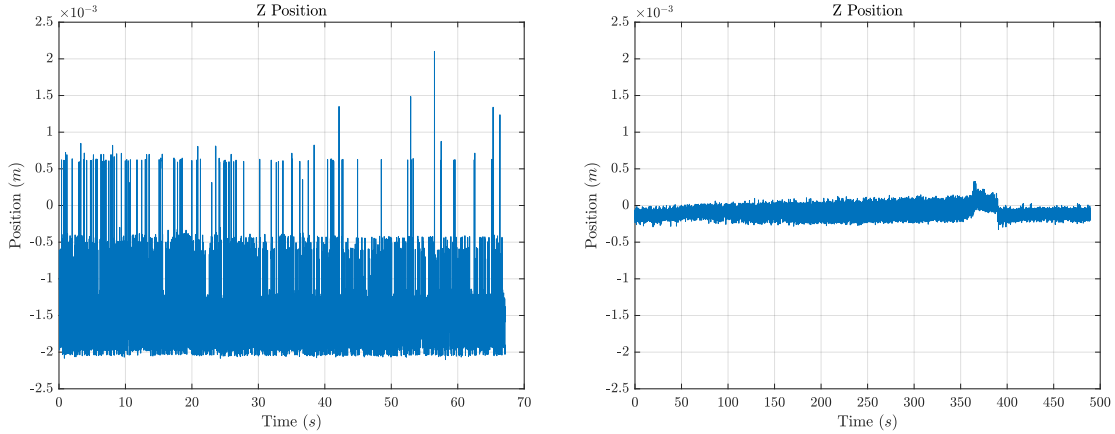

Fig. 5.9: Position in the navigation Z direction during baseline test for narrow reflector placement (left) and wide reflector placement (right)

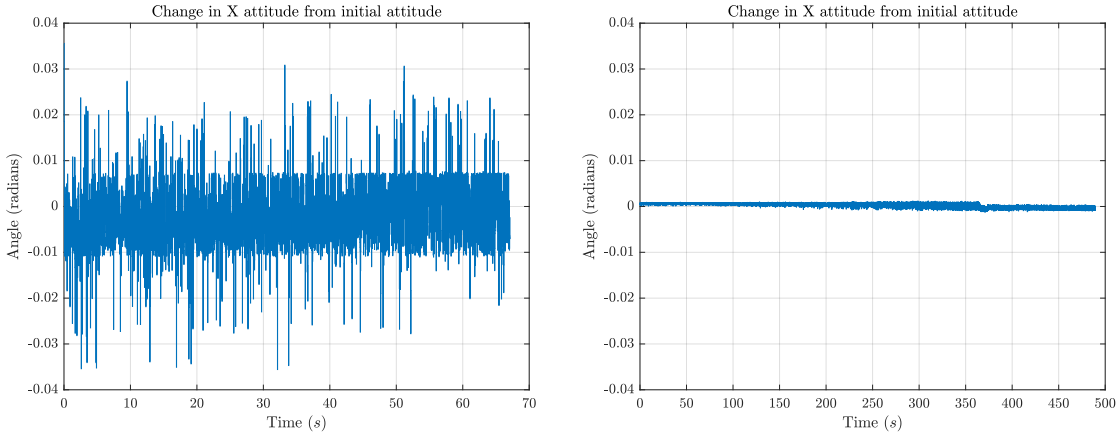

Fig. 5.10: Attitude rotation about the navigation X axis during baseline test for narrow reflector placement (left) and wide reflector placement (right)

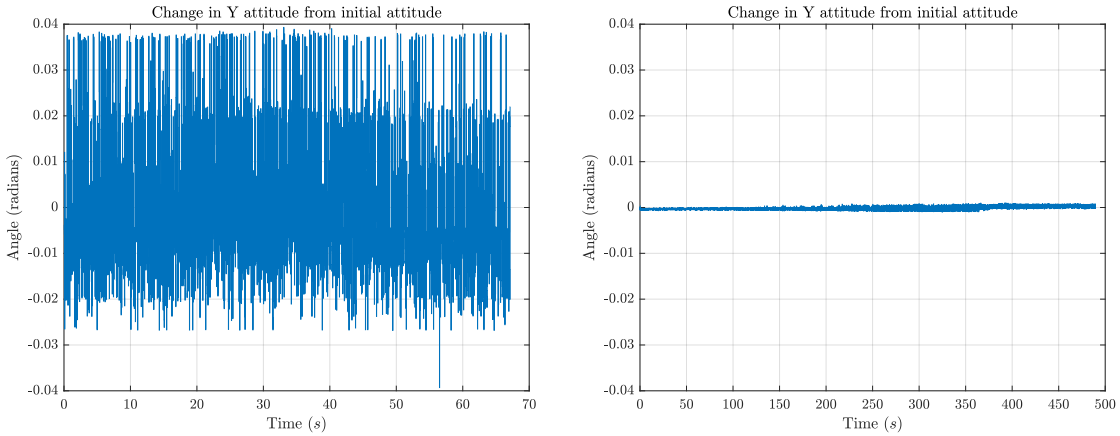

Fig. 5.11: Attitude rotation about the navigation Y axis during baseline test for narrow reflector placement (left) and wide reflector placement (right)

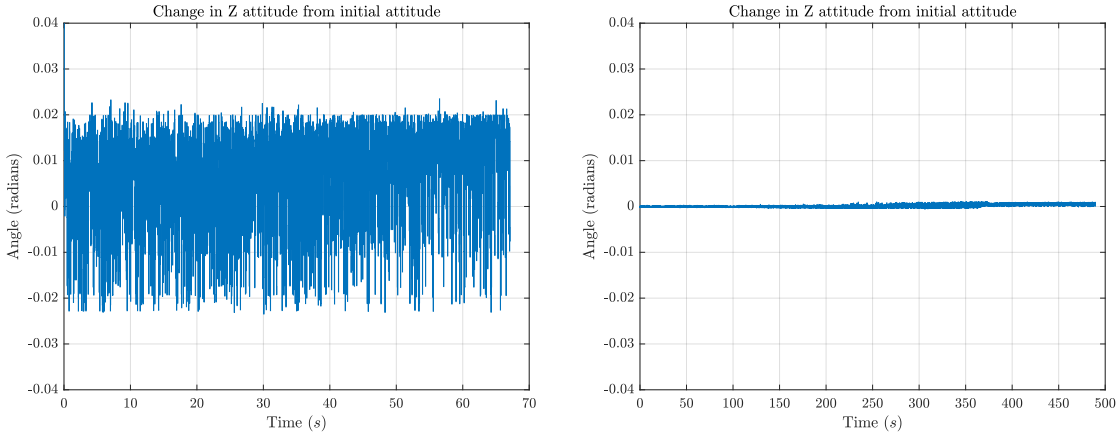

Fig. 5.12: Attitude rotation about the navigation Z axis during baseline test for narrow reflector placement (left) and wide reflector placement (right)

The error values in Table 5.1 are the standard deviations of values shown in Figures 5.7 to 5.12. It can be seen that placing the reflective markers for the motion capture system in a wider configuration reduced the standard deviation of position measurements by around a factor of 10 and attitude mesurements by up to a factor of 100. It may be that if the test for the narrower configuration were allowed to run for a longer time, it might have had an even larger range of measurements. An interesting feature in these plots is that for the wide reflector placement, the measurements were shown to shift at about 370 seconds. The standard deviations of the measurements were also not constant over time. This indicates that these measurements may not be sufficiently zero-mean and white to be used in a Kalman filter.

Measurement	Narrow Reflector Placement	Wide Reflector Placement
Position X (mm)	1.345	0.113
Position Y (mm)	0.635	0.080
Position Z (mm)	0.514	0.067
Attitude X (radians)	0.00786	0.00047
Attitude Y (radians)	0.01561	0.00036
Attitude Z (radians)	0.01165	0.00030

Table 5.1: Standard deviation of position and attitude measurements for the two test configurations of motion capture reflective markers while the UAV was sitting flat on the ground

CHAPTER 6
CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this thesis, a system for GPS-denied navigation of a UAV was discussed. Previous research that has contributed to this area of interest was reviewed along with the need for further research. The research in this thesis was aimed at analyzing the type of configuration required to maintain acceptable amounts of state estimation error using a system that fuses IMU measurements with LOS measurements to self-describing fiducials through an EKF. Specifically, the effects of IMU grade and fiducial spacing along a UAV's trajectory as well as other parameters such as height above fiducials and frequency at which camera images were processed was examined. This was done by developing the equations for the EKF, validating those equations through Monte Carlo simulations, and comparing the effects of IMU grades, fiducial spacings, UAV trajectory height above the fiducials, and image processing frequency in simulation.

In the simulation data, it can be seen that pairing IMU measurements with LOS measurements to fiducials that have self-describing positions is an effective way for a UAV to estimate its dynamic states, particularly its position. With a fiducial spacing of 10 meters apart, fiducials 15 meters below the UAV's trajectory, and an image processing frequency of 5 Hz, 3-σ values for the estimation error of position can be as small as around 0.2 meters regardless of the grade of IMU used. As fiducials are spaced further apart, commercial-grade IMUs begin to be significantly less effective than tactical and navigation-grade IMUs. However, there is not as large of a difference between the tactical and navigation-grade IMUs. The ideal setup required to obtain desired position estimation would need to be determined based on mission requirements.

The simulation data also showed that the height above the fiducials can affect how much

the fiducial measurements help because as the fiducials are further away, there is more time for the camera to capture images of each fiducial, and it is more likely that the camera would capture more than one fiducial at a given moment. However, the further away a fiducial is, the less sensitive the EKF is to the measurement, and the less it corrects the state estimation errors. For a setup utilizing a tactical-grade IMU and processing images at 5Hz, the north position estimation error varied by approximately 0.27 meters. The frequency of image sampling had an even greater effect. With a tactical-grade IMU and fiducials placed 15 m below the trajectory, the north position estimation error varied by approximately 0.74 meters as the time between image samples was varied between 0.05 seconds and 1 second.

For the purpose of implementing the developed EKF in hardware, ArUco markers were suggested as a viable option, and a means of reducing ArUco markers to a single point was developed and shown to work. A motion capture system was proposed for the purpose of gathering truth position and attitude states for the UAV to compare to estimated states from the EKF. The noise associated with the position and attiude of a UAV sitting still in a motion capture system was observed. The reduction of this noise caused by adding extenders for expanding the radius of the reflective markers in the motion capture area was analyzed.

This study has taken the idea of using self-describing fiducials to aid in GPS-denied navigation of UAVs and simulated this scenario using an Extended Kalman Filter in multiple runs of Monte Carlo simulations. This has shown the potential effectiveness of various fiducial densities and IMU grades for the application of this strategy. This provides a viable alternative to other approaches to GPS-denied navigation such as Simultaneous Location and Mapping and pure Visual Odometry. Preparations have been made to begin implementation of this system in hardware.

6.2 Future Work

Future work can be done in several areas with this study. On the simulation side, one option would be to determine the most effective way for a UAV to detect fiducials. It would be interesting to evaluate the effect of varying the camera-to-body frame rotation. It

would also be interesting to analyze the effects of various camera specifications on the state estimation capability. It would also be of value to analyze the effect of not just spacing between fiducials, but the amount of time between when a fiducial is seen.

It could also be a beneficial study to determine the effectiveness of processing multiple fiducials as part of a single measurement. The study in this research treated each fiducial as an individual measurement. This may not be the most effective approach. It might be more effective to form a type of triangulation whenever more than one fiducial is in view and use this as the measurement instead.

Future work can also be done by implementing this system fully in hardware and comparing the results obtained in simulation to results obtained in a hardware system. It could be useful to analyze the amount of error present in the motion capture system when a rigid body is moving around in the captured area and not just sitting still in one position as was analyzed in this study. This would show how much the motion capture data can be relied on as truth data for comparison to the states estimated by the EKF.

REFERENCES

[1] J. K. Stolaroff, "The Need for a Life Cycle Assessment of Drone-Based Commercial Package Delivery," Tech. Rep. LLNL-TR-652316, 1129145, Mar. 2014. [Online]. Available: http://www.osti.gov/servlets/purl/1129145/

[2] S. X. Fang, S. O'Young, and L. Rolland, "Development of Small UAS Beyond-Visual-Line-of-Sight (BVLOS) Flight Operations: System Requirements and Procedures," *Drones*, vol. 2, no. 2, p. 13, Jun. 2018. [Online]. Available: https://www.mdpi.com/2504-446X/2/2/13

[3] R. B. Langley, "Dilution of precision," *GPS world*, vol. 10, no. 5, pp. 52–59, 1999. [Online]. Available: http://131.202.94.44/papers.pdf/gpsworld.may99.pdf

[4] A. El-Rabbany, *Introduction to GPS: The Global Positioning System*. Artech House, 2002, google-Books-ID: U2JmghrrB8cC.

[5] E. Kaplan and C. Hegarty, *Understanding GPS: Principles and Applications*. Artech House, 2005.

[6] M. G. Wing, A. Eklund, and L. D. Kellogg, "Consumer-Grade Global Positioning System (GPS) Accuracy and Reliability," *Journal of Forestry*, vol. 103, no. 4, pp. 169–173, Jun. 2005, publisher: Oxford Academic. [Online]. Available: https://academic.oup.com/jof/article/103/4/169/4598618

[7] M. Modsching and R. Kramer, "Field trial on GPS Accuracy in a medium size city: The influence of built- up," p. 10, 2006.

[8] M. E. Davidson, R. S. Christensen, and R. C. Robinson, "Navigation Using Self-Describing Fiducials," US Patent US20 170 323 129A1, Nov., 2017. [Online]. Available: https://patents.google.com/patent/US20170323129/en?oq=randall+christensen

[9] "VTOL FixedWing | VTOL." [Online]. Available: https://cad.onshape.com/documents/a19d8d6cf38b3c2a78612528/w/d6a6399219628700f8e37bb4/e/d647e76864c3a3c1150321b6

[10] "Qr Code," Oct. 2017. [Online]. Available: https://pixabay.com/vectors/qr-code-bar-code-code-2816041/

[11] "Modern city, town street." [Online]. Available: https://www.freepik.com/free-vector/modern-city-town-street_4393644.htm

[12] G. Chowdhary, E. N. Johnson, D. Magree, A. Wu, and A. Shein, "GPS-denied Indoor and Outdoor Monocular Vision Aided Navigation and Control of Unmanned Aircraft," *Journal of Field Robotics*, vol. 30, no. 3, pp. 415–438, 2013. [Online]. Available: https://www.onlinelibrary.wiley.com/doi/abs/10.1002/rob.21454

[13] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-SLAM–based navigation for autonomous micro helicopters in GPS-denied environments - Weiss - 2011 - Journal of Field Robotics - Wiley Online Library," *Journal of Field Robotics*, vol. 28, no. 6, pp. 854–874, Oct. 2011. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1002/rob.20412

[14] M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart, "Vision based MAV navigation in unknown and unstructured environments," in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 21–28, iSSN: 1050-4729.

[15] J. Civera, A. J. Davison, and J. M. M. Montiel, "Inverse Depth Parametrization for Monocular SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, Oct. 2008, conference Name: IEEE Transactions on Robotics.

[16] J. Kim and S. Sukkarieh, "Autonomous airborne navigation in unknown terrain environments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 3, pp. 1031–1045, Jul. 2004, conference Name: IEEE Transactions on Aerospace and Electronic Systems.

[17] R. C. Leishman, T. W. McLain, and R. W. Beard, "Relative Navigation Approach for Vision-Based Aerial GPS-Denied Navigation," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1, pp. 97–111, Apr. 2014. [Online]. Available: https://doi.org/10.1007/s10846-013-9914-7

[18] D. O. Wheeler, D. P. Koch, J. S. Jackson, T. W. McLain, and R. W. Beard, "Relative Navigation: A Keyframe-Based Approach for Observable GPS-Degraded Navigation," *IEEE Control Systems Magazine*, vol. 38, no. 4, pp. 30–48, Aug. 2018, conference Name: IEEE Control Systems Magazine.

[19] M. J. Veth, "Navigation Using Images, A Survey of Techniques," *NAVIGATION*, vol. 58, no. 2, pp. 127–139, 2011, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.2161-4296.2011.tb01796.x. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2161-4296.2011.tb01796.x

[20] D. P. Koch, D. O. Wheeler, R. W. Beard, T. W. McLain, and K. M. Brink, "Relative multiplicative extended Kalman filter for observable GPS-denied navigation," *The International Journal of Robotics Research*, vol. 39, no. 9, Jun. 2020. [Online]. Available: https://journals.sagepub.com/doi/full/10.1177/0278364920903094

[21] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, Jun. 2004, pp. I–I, iSSN: 1063-6919.

[22] A. Al-Kaff, D. Martín, F. García, A. d. l. Escalera, and J. María Armingol, "Survey of computer vision algorithms and applications for unmanned aerial vehicles," *Expert Systems with Applications*, vol. 92, pp. 447–463, Feb. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417417306395

[23] Chaolei Wang, Tianmiao Wang, Jianhong Liang, Yang Chen, and Yongliang Wu, "Monocular vision and IMU based navigation for a small unmanned helicopter," in *2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Jul. 2012, pp. 1694–1699.

[24] D. Scaramuzza and F. Fraundorfer, "Visual Odometry [Tutorial]," *IEEE Robotics Automation Magazine*, vol. 18, no. 4, pp. 80–92, Dec. 2011, conference Name: IEEE Robotics Automation Magazine.

[25] M. G. Wing, A. Eklund, and L. D. Kellogg, "Consumer-Grade Global Positioning System (GPS) Accuracy and Reliability," *Journal of Forestry*, vol. 103, no. 4, pp. 169–173, Jun. 2005. [Online]. Available: https://academic.oup.com/jof/article/103/4/169/4598618

[26] B. Yun, K. Peng, and B. M. Chen, "Enhancement of GPS Signals for Automatic Control of a UAV Helicopter System," in *2007 IEEE International Conference on Control and Automation*, May 2007, pp. 1185–1189, iSSN: 1948-3457.

[27] J.-H. Kim and S. Sukkarieh, "Flight Test Results of GPS/INS Navigation Loop for an Autonomous Unmanned Aerial Vehicle (UAV)," Sep. 2002, pp. 510–517. [Online]. Available: http://www.ion.org/publications/abstract.cfm?jp=p&articleID=2052

[28] M. Rieke, T. Foerster, J. Geipel, and T. Prinz, "HIGH-PRECISION POSITIONING AND REAL-TIME DATA PROCESSING OF UAV-SYSTEMS," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVIII-1/C22, pp. 119–124, Sep. 2012. [Online]. Available: http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XXXVIII-1-C22/119/2011/

[29] A. Nemra and N. Aouf, "Robust INS/GPS Sensor Fusion for UAV Localization Using SDRE Nonlinear Filtering," *IEEE Sensors Journal*, vol. 10, no. 4, pp. 789–798, Apr. 2010, conference Name: IEEE Sensors Journal.

[30] C. Y. C.-S. Yoo and I. A. I.-K. Ahn, "Low cost GPS/INS sensor fusion system for UAV navigation," in *Digital Avionics Systems Conference, 2003. DASC '03. The 22nd*, vol. 2, Oct. 2003, pp. 8.A.1–8.1–9 vol.2.

[31] M. Orsag, C. Korpela, and P. Oh, "Modeling and Control of MM-UAV: Mobile Manipulating Unmanned Aerial Vehicle," *Journal of Intelligent & Robotic Systems*, vol. 69, no. 1, pp. 227–240, Jan. 2013. [Online]. Available: https://doi.org/10.1007/s10846-012-9723-4

[32] J. J. Troy, C. A. Erignac, and P. Murray, "Haptics-Enabled UAV Teleoperation Using Motion Capture Systems," *Journal of Computing and Information Science in Engineering*, vol. 9, no. 1, Mar. 2009, publisher: American Society of Mechanical Engineers Digital Collection. [Online]. Available: https://asmedigitalcollection.asme.org/computingengineering/article/9/1/011003/398369/Haptics-Enabled-UAV-Teleoperation-Using-Motion

[33] E. J. Rodríguez-Seda, J. J. Troy, C. A. Erignac, P. Murray, D. M. Stipanovic, and M. W. Spong, "Bilateral Teleoperation of Multiple Mobile Agents: Coordinated Motion and Collision Avoidance," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 4, pp. 984–992, Jul. 2010, conference Name: IEEE Transactions on Control Systems Technology.

[34] K. Chang, J. H. Kim, S. A. Wilkerson, and S. A. Gadsden, "Motion Capture Control of a Nano Quadrotor," p. 6, 2017.

[35] L. L. Gomes, L. Leal, T. R. Oliveira, J. P. V. S. Cunha, and T. C. Revoredo, "Unmanned Quadcopter Control Using a Motion Capture System," *IEEE Latin America Transactions*, vol. 14, no. 8, pp. 3606–3613, Aug. 2016, conference Name: IEEE Latin America Transactions.

[36] A. M. Aurand, J. S. Dufour, and W. S. Marras, "Accuracy map of an optical motion capture system with 42 or 21 cameras in a large measurement volume," *Journal of Biomechanics*, vol. 58, pp. 237–240, Jun. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021929017302580

[37] A. J. Smith, E. Wang, J. C. LaCombe, and T. Fields, "Low Cost Alternative to Motion Capture Systems for Indoor Flight Testing Using On-board Computer Vision," in *AIAA Atmospheric Flight Mechanics Conference*. American Institute of Aeronautics and Astronautics, 2016, _eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2016-1754. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2016-1754

[38] C. H. Tseng and J.-S. Yen, "Enhanced Gaussian mixture model of RSSI purification for indoor positioning," *Journal of Systems Architecture*, vol. 81, pp. 1–6, Nov. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1383762117301066

[39] H. Al-Moukhles, I. Abdel-Qader, and A. K. Abed, "Multiple-Fingerprints technique for WLAN and compressive sensing based IPS," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan. 2018, pp. 707–713.

[40] B. Han and L. Zhao, "An indoor positioning algorithm based on Wi-Fi fingerprint and inertial navigation system," in *2017 36th Chinese Control Conference (CCC)*, Jul. 2017, pp. 6067–6072, iSSN: 1934-1768.

[41] W. Jung and Y. Ko, "IEEE 802.11 ac based client centered indoor positioning system," in *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, May 2016, pp. 1–4.

[42] V. Cantón Paterna, A. Calveras Augé, J. Paradells Aspas, and M. A. Pérez Bullones, "A Bluetooth Low Energy Indoor Positioning System with Channel Diversity, Weighted Trilateration and Kalman Filtering," *Sensors*, vol. 17, no. 12, p. 2927, Dec. 2017, number: 12 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/17/12/2927

[43] L. Mainetti, L. Patrono, and I. Sergi, "A survey on indoor positioning systems," in *2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Sep. 2014, pp. 111–120.

[44] G. Casati, M. Longhi, D. Latini, F. Carbone, S. Amendola, F. D. Frate, G. Schiavon, and G. Marrocco, "The Interrogation Footprint of RFID-UAV: Electromagnetic Modeling and Experimentations," *IEEE Journal of Radio Frequency Identification*, vol. 1, no. 2, pp. 155–162, Jun. 2017, conference Name: IEEE Journal of Radio Frequency Identification.

[45] M. Longhi, Z. Taylor, M. Popović, J. Nieto, G. Marrocco, and R. Siegwart, "RFID-Based Localization for Greenhouses Monitoring Using MAVs," in *2018 IEEE-APS Topical Conference on Antennas and Propagation in Wireless Communications (APWC)*, Sep. 2018, pp. 905–908.

[46] M. Longhi and G. Marrocco, "Ubiquitous Flying Sensor Antennas: Radiofrequency Identification Meets Micro Drones," *IEEE Journal of Radio Frequency Identification*, vol. 1, no. 4, pp. 291–299, Dec. 2017, conference Name: IEEE Journal of Radio Frequency Identification.

[47] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 3400–3407.

[48] T. Gerratt, A. Strate, and R. Christensen, "Navigation Error Sensitivity of Autonomous Carrier Landing Systems in GPS-denied Environments,," Portland, Oregon, Apr. 2020, pp. 81–90. [Online]. Available: https://www.ion.org/publications/abstract.cfm?articleID=17444

[49] P. H. Nguyen, K. W. Kim, Y. W. Lee, and K. R. Park, "Remote Marker-Based Tracking for UAV Landing Using Visible-Light Camera Sensor," *Sensors*, vol. 17, no. 9, p. 1987, Sep. 2017, number: 9 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/17/9/1987

[50] 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

[51] N. Kahani, A. Heins, W. Zhao, M. Nahangi, B. Mccabe, and A. Schoellig, "Improved Tag-based Indoor Localization of UAVs Using Extended Kalman Filter," 2019. [Online]. Available: https://www.researchgate.net/publication/333633015_Improved_Tag-based_Indoor_Localization_of_UAVs_Using_Extended_Kalman_Filter

[52] M. Nahangi, A. Heins, B. Mccabe, and A. Schoellig, "Automated Localization of UAVs in GPS-Denied Indoor Construction Environments Using Fiducial Markers," Jul. 2018.

[53] N. Kayhani, B. McCabe, A. Abdelaal, A. Heins, and A. Schoellig, Angela P., "Tag-based Indoor Localization of UAVs in Construction Environments: Opportunities and Challenges in Practice." [Online]. Available: http://www.dynsyslab.org/wp-content/papercite-data/pdf/kayhani-crc20.pdf

[54] A. Andziulis, D. Drungilas, V. Glazko, and E. Kiseliovas, "Resource Saving Approach of Visual Tracking Fiducial Marker Recognition for Unmanned Aerial Vehicle," *Advances in Electrical and Electronic Engineering*, vol. 13, no. 4, pp. 359 – 366, Nov. 2015. [Online]. Available: http://advances.utc.sk/index.php/AEEE/article/view/1492

[55] V. Mondéjar-Guerra, S. Garrido-Jurado, R. Muñoz-Salinas, M. J. Marín-Jiménez, and R. Medina-Carnicer, "Robust identification of fiducial markers in challenging conditions," *Expert Systems with Applications*, vol. 93, pp. 336–345, Mar. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417417307145

[56] J. Wang and E. Olson, "AprilTag 2: Efficient and robust fiducial detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2016, pp. 4193–4198.

[57] M. Krogius, A. Haggenmiller, and E. Olson, "Flexible Layouts for Fiducial Tags," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[58] M. Fiala, "ARTag, a fiducial marker system using digital techniques," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, Jun. 2005, pp. 590–596 vol. 2.

[59] ——, "Designing Highly Reliable Fiducial Markers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1317–1324, Jul. 2010, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[60] K. Shabalina, A. Sagitov, H. Li, and E. Magid, "Comparing Fiducial Marker Systems Occlusion Resilience through a Robot Eye," in *2017 10th International Conference on Developments in eSystems Engineering (DeSE)*, Jun. 2017, pp. 273–278, iSSN: 2161-1343.

[61] L. Cavanini, G. Cimini, F. Ferracuti, A. Freddi, G. Ippoliti, A. Monteriù, and F. Verdini, "A QR-code localization system for mobile robots: Application to smart wheelchairs," in *2017 European Conference on Mobile Robots (ECMR)*, Sep. 2017, pp. 1–6.

[62] A. Censi, J. Strubel, C. Brandli, T. Delbruck, and D. Scaramuzza, "Low-latency localization by active LED markers tracking using a dynamic vision sensor," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2013, pp. 891–898, iSSN: 2153-0866.

[63] M. Faessler, E. Mueggler, K. Schwabe, and D. Scaramuzza, "A monocular pose estimation system based on infrared LEDs," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 907–913, iSSN: 1050-4729.

[64] D. Claus and A. W. Fitzgibbon, "Reliable Fiducial Detection in Natural Scenes," in *Computer Vision - ECCV 2004*, ser. Lecture Notes in Computer Science, T. Pajdla and J. Matas, Eds. Berlin, Heidelberg: Springer, 2004, pp. 469–480.

[65] D. B. d. S. Cesar, C. Gaudig, M. Fritsche, M. A. d. Reis, and F. Kirchner, "An evaluation of artificial fiducial markers in underwater environments - IEEE Conference Publication," 2015. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7271491

[66] M. S. Grewal, A. P. Andrews, and C. G. Bartone, *Global Navigation Satellite Systems, Inertial Navigation, and Integration*. John Wiley & Sons, Jan. 2020, google-Books-ID: ppjDDwAAQBAJ.

[67] J. Farrell, *Aided Navigation: GPS with High Rate Sensors*, 1st ed. USA: McGraw-Hill, Inc., 2008.

[68] P. Savage, *Strapdown analytics*. Maple Plain, Minn.: Strapdown Associates, 2000. [Online]. Available: http://strapdownassociates.com/sdanal.brochure.pdf

[69] P. S. Maybeck, "5.4 Statistics of processes within the filter structure," in *Stochastic models, estimation and control*, ser. Mathematics in science and engineering. New York: Academic Press, 1979, no. v, p. 229.

[70] "AprilRobotics/apriltag_ros," Sep. 2020, original-date: 2017-12-14T22:04:09Z. [Online]. Available: https://github.com/AprilRobotics/apriltag_ros

[71] "ros-drivers/zbar_ros," May 2020, original-date: 2015-01-10T18:51:55Z. [Online]. Available: https://github.com/ros-drivers/zbar_ros

[72] "UbiquityRobotics/fiducials," Sep. 2020, original-date: 2015-02-15T06:11:15Z. [Online]. Available: https://github.com/UbiquityRobotics/fiducials

APPENDICES

APPENDIX A

Linearization of State Equations

For the dynamics of a UAV, the truth and design models are assumed to be equivalent. Here, the truth model in Equation 3.4 is linearized using the perturbation method of linearization about the estimated states.

Equation 3.8 shows the design model dynamics and is duplicated below:

$$\dot{\underline{x}} = \begin{bmatrix} \dot{\underline{p}}^n \\ \dot{\underline{v}}^n \\ \dot{q}_b^n \\ \dot{\underline{b}}_a \\ \dot{\underline{b}}_g \\ \dot{q}_c^b \end{bmatrix} = \begin{bmatrix} \underline{v}^n \\ R_b^n \, \underline{\nu}^b + \underline{g}^n \\ \frac{1}{2} \, q_b^n \otimes \begin{bmatrix} 0 \\ \underline{\omega}^b \end{bmatrix} \\ -\frac{1}{\tau_a} \, \underline{b}_a + \underline{n}_a \\ -\frac{1}{\tau_g} \, \underline{b}_g + \underline{n}_g \\ \begin{bmatrix} 0 \; 0 \; 0 \; 0 \end{bmatrix}^T \end{bmatrix}$$

These are the equations that will be linearized about $\hat{\underline{x}}$.

A.1 Position Linearization

The position vector derivative is

$$\dot{\underline{p}}^n = \underline{v}^n \tag{A.1}$$

Substituting definitions of $\underline{p}$ and $\underline{v}$ from Equation 3.4 yields

$$\dot{\hat{\underline{p}}}^n + \delta\dot{\underline{p}}^n = \hat{\underline{v}}^n + \delta v^n \tag{A.2}$$

Canceling equal terms as found in Equation 3.9 from both sides gives the linearized equation for the position vector.

$$\delta \dot{\underline{p}}^n = \delta v^n \tag{A.3}$$

A.2 Velocity Linearization

The velocity vector derivative is

$$\dot{\underline{v}} = R_b^n \, \underline{\nu}^b + \underline{g}^n \tag{A.4}$$

Substituting definitions from Equation 3.4 yields

$$\dot{\hat{\underline{v}}}^n + \delta \dot{\underline{v}}^n = R_b^n \, \underline{\nu}^b + \underline{g}^n \tag{A.5}$$

Substitution of the definition of the rotation vector $R_b^n \equiv (I - \delta \underline{\theta}_b^n \times) \hat{R}_b^n$ yields

$$\dot{\hat{\underline{v}}}^n + \delta \dot{\underline{v}}^n = \left[(I - \delta \underline{\theta}_b^n \times) \, \hat{R}_b^n \right] \underline{\nu}^b + \underline{g}^n \tag{A.6}$$

Substitution of the definition of $\underline{\nu}^b$ from Equation 3.10 and the definition of $\underline{b}_a$ from Equation 3.4 yields

$$\dot{\hat{\underline{v}}}^n + \delta \dot{\underline{v}}^n = \left[(I - \delta \underline{\theta}_b^n \times) \, \hat{R}_b^n \right] \left(\tilde{\underline{\nu}}^b - \hat{\underline{b}}_a - \delta \underline{b}_a - \underline{n}_\nu \right) + \underline{g}^n \tag{A.7}$$

Expanding yields

$$\dot{\hat{\underline{v}}}^n + \delta \dot{\underline{v}}^n = \hat{R}_b^n \left(\tilde{\underline{\nu}}^b - \hat{\underline{b}}_a - \delta \underline{b}_a - \underline{n}_\nu \right) - (\delta \underline{\theta}_b^n \times) \, \hat{R}_b^n \left(\tilde{\underline{\nu}}^b - \hat{\underline{b}}_a - \delta \underline{b}_a - \underline{n}_\nu \right) + \underline{g}^n \tag{A.8}$$

Canceling the definition of $\dot{\hat{\underline{v}}}^n$ from Equation 3.9 yields

$$\delta \dot{\underline{v}}^n = \hat{R}_b^n \left(-\delta \underline{b}_a - \underline{n}_\nu \right) - (\delta \underline{\theta}_b^n \times) \, \hat{R}_b^n \left(\tilde{\underline{\nu}}^b - \hat{\underline{b}}_a - \delta \underline{b}_a - \underline{n}_\nu \right) \tag{A.9}$$

Reversing the order of the cross product yields

$$\delta\dot{\underline{v}}^n = \hat{R}_b^n\left(-\delta\underline{b}_a - \underline{n}_\nu\right) + \left[\hat{R}_b^n\left(\tilde{\underline{\nu}}^b - \hat{\underline{b}}_a - \delta\underline{b}_a - \underline{n}_\nu\right)\right] \times \delta\underline{\theta}_b^n \tag{A.10}$$

Taking away second-order terms yields

$$\delta\dot{\underline{v}}^n = \hat{R}_b^n\left(-\delta\underline{b}_a - \underline{n}_\nu\right) + \left[\hat{R}_b^n\left(\tilde{\underline{\nu}}^b - \hat{\underline{b}}_a\right)\right] \times \delta\underline{\theta}_b^n \tag{A.11}$$

Rearranging of terms give the final form

$$\delta\dot{\underline{v}}^n = \left[\hat{R}_b^n\left(\tilde{\underline{\nu}}^b - \hat{\underline{b}}_a\right)\right] \times \delta\underline{\theta}_b^n - \hat{R}_b^n\delta\underline{b}_a - \hat{R}_b^n\underline{n}_\nu \tag{A.12}$$

A.3 UAV Body Frame Linearization

The body frame to NED quaternion derivative is defined as

$$\dot{q}_b^n = \frac{1}{2}\, q_b^n \otimes \begin{bmatrix} 0 \\ \\ \underline{\omega}^b \end{bmatrix} \tag{A.13}$$

Letting $\delta q \equiv \begin{bmatrix} 1 \\ \\ -\frac{1}{2}\delta\underline{\theta}_b^n \end{bmatrix}$ and substituting in the definition of q_b^n from equation 3.4 yields

$$\frac{d}{dt}(\delta q \otimes \hat{q}_b^n) = \frac{1}{2}\delta q \otimes \hat{q}_b^n \otimes \begin{bmatrix} 0 \\ \\ \underline{\omega}^b \end{bmatrix} \tag{A.14}$$

The angular velocity $\underline{\omega}^b$ is defined as $\underline{\omega}^b \equiv \tilde{\underline{\omega}}^b - \hat{\underline{b}}_g - \underline{n}_\omega$. Substituting this and the definition of $\underline{b}_g$ from Equation 3.4 yields

$$\frac{d}{dt}(\delta q \otimes \hat{q}_b^n) = \frac{1}{2}\delta q \otimes \hat{q}_b^n \otimes \begin{bmatrix} 0 \\ \\ \tilde{\underline{\omega}}^b - \hat{\underline{b}}_g - \delta\underline{b}_g - \underline{n}_\omega \end{bmatrix} \tag{A.15}$$

Expanding the derivative via the multiplication rule yields

$$\delta\dot{q} \otimes \hat{q}_b^n + \delta q \otimes \dot{\hat{q}}_b^n = \frac{1}{2}\delta q \otimes \hat{q}_b^n \otimes \begin{bmatrix} 0 \\ \underline{\tilde{\omega}}^b - \underline{\hat{b}}_g - \delta\underline{b}_g - \underline{n}_\omega \end{bmatrix} \tag{A.16}$$

The matrix on the right hand side can be split into two parts,

$$\delta\dot{q} \otimes \hat{q}_b^n + \delta q \otimes \dot{\hat{q}}_b^n = \frac{1}{2}\delta q \otimes \hat{q}_b^n \otimes \left(\begin{bmatrix} 0 \\ \underline{\tilde{\omega}}^b - \underline{\hat{b}}_g \end{bmatrix} + \begin{bmatrix} 0 \\ -\delta\underline{b}_g - \underline{n}_\omega \end{bmatrix} \right) \tag{A.17}$$

and the quaternion multiplication can be distributed to yield

$$\delta\dot{q} \otimes \hat{q}_b^n + \delta q \otimes \dot{\hat{q}}_b^n = \frac{1}{2}\delta q \otimes \hat{q}_b^n \otimes \begin{bmatrix} 0 \\ \underline{\tilde{\omega}}^b - \underline{\hat{b}}_g \end{bmatrix} + \frac{1}{2}\delta q \otimes \hat{q}_b^n \otimes \begin{bmatrix} 0 \\ -\delta\underline{b}_g - \underline{n}_\omega \end{bmatrix} \tag{A.18}$$

Substituting the definition of $\dot{\hat{q}}_b^n$ from equation 3.9 into the first term on the right hand side yields

$$\delta\dot{q} \otimes \hat{q}_b^n + \delta q \otimes \dot{\hat{q}}_b^n = \delta q \otimes \dot{\hat{q}}_b^n + \frac{1}{2}\delta q \otimes \hat{q}_b^n \otimes \begin{bmatrix} 0 \\ -\delta\underline{b}_g - \underline{n}_\omega \end{bmatrix} \tag{A.19}$$

Canceling like terms yields

$$\delta\dot{q} \otimes \hat{q}_b^n = \frac{1}{2}\delta q \otimes \hat{q}_b^n \otimes \begin{bmatrix} 0 \\ -\delta\underline{b}_g - \underline{n}_\omega \end{bmatrix} \tag{A.20}$$

Right hand quaternion multiplication by $(\hat{q}_b^n)^*$ yields

$$\delta\dot{q} \otimes \hat{q}_b^n \otimes (\hat{q}_b^n)^* = \delta\dot{q} = \frac{1}{2}\delta q \otimes \hat{q}_b^n \otimes \begin{bmatrix} 0 \\ -\delta\underline{b}_g - \underline{n}_\omega \end{bmatrix} \otimes (\hat{q}_b^n)^* \tag{A.21}$$

A quaternion and its quaternion conjugate pre- and post-multiplied by something is equivalent to multiplying it by a corresponding rotation vector. Applying this definition,

$\delta\dot{q}$ becomes

$$\delta\dot{q} = \frac{1}{2}\delta q \otimes \begin{bmatrix} 0 \\ -\hat{R}_b^n \delta\underline{b}_g - \hat{R}_b^n \underline{n}_\omega \end{bmatrix} \tag{A.22}$$

Substituting $\delta q \equiv \begin{bmatrix} 1 \\ -\frac{1}{2}\delta\underline{\theta}_b^n \end{bmatrix}$ yields

$$\begin{bmatrix} 1 \\ -\frac{1}{2}\delta\underline{\dot{\theta}}_b^n \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 \\ -\frac{1}{2}\delta\underline{\theta}_b^n \end{bmatrix} \otimes \begin{bmatrix} 0 \\ -\hat{R}_b^n \delta\underline{b}_g - \hat{R}_b^n \underline{n}_\omega \end{bmatrix} \tag{A.23}$$

Performing the quaternion multiplication yields

$$\begin{bmatrix} 1 \\ -\frac{1}{2}\delta\underline{\dot{\theta}}_b^n \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -\frac{1}{2}\delta\underline{\theta}_b^n \cdot (-\hat{R}_b^n \delta\underline{b}_g - \hat{R}_b^n \underline{n}_\omega) \\ -\frac{1}{2}\delta\underline{\theta}_b^n \times (-\hat{R}_b^n \delta\underline{b}_g - \hat{R}_b^n \underline{n}_\omega) - \hat{R}_b^n \delta\underline{b}_g - \hat{R}_b^n \underline{n}_\omega \end{bmatrix} \tag{A.24}$$

Factoring the 1/2 yields

$$\frac{1}{2}\begin{bmatrix} 2 \\ -\delta\underline{\dot{\theta}}_b^n \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -\frac{1}{2}\delta\underline{\theta}_b^n \cdot (-\hat{R}_b^n \delta\underline{b}_g - \hat{R}_b^n \underline{n}_\omega) \\ -\frac{1}{2}\delta\underline{\theta}_b^n \times (-\hat{R}_b^n \delta\underline{b}_g - \hat{R}_b^n \underline{n}_\omega) - \hat{R}_b^n \delta\underline{b}_g - \hat{R}_b^n \underline{n}_\omega \end{bmatrix} \tag{A.25}$$

Canceling the 1/2 yields

$$\begin{bmatrix} 2 \\ -\delta\underline{\dot{\theta}}_b^n \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}\delta\underline{\theta}_b^n \cdot (-\hat{R}_b^n \delta\underline{b}_g - \hat{R}_b^n \underline{n}_\omega) \\ -\frac{1}{2}\delta\underline{\theta}_b^n \times (-\hat{R}_b^n \delta\underline{b}_g - \hat{R}_b^n \underline{n}_\omega) - \hat{R}_b^n \delta\underline{b}_g - \hat{R}_b^n \underline{n}_\omega \end{bmatrix} \tag{A.26}$$

Extracting the second row of the matrix equality yields

$$-\delta\underline{\dot{\theta}}_b^n = -\frac{1}{2}\delta\underline{\theta}_b^n \times (-\hat{R}_b^n \delta\underline{b}_g - \hat{R}_b^n \underline{n}_\omega) - \hat{R}_b^n \delta\underline{b}_g - \hat{R}_b^n \underline{n}_\omega \tag{A.27}$$

Canceling second-order terms and rearranging yields the final form

$$\delta\underline{\dot{\theta}}_b^n = \hat{R}_b^n \delta\underline{b}_g + \hat{R}_b^n \underline{n}_\omega \tag{A.28}$$

A.4 Linearization of Biases and Camera-to-Body Quaternion

Start with the accelerometer bias derivative

$$\dot{\underline{b}}_a = -\frac{1}{\tau_a}\,\underline{b}_a + \underline{n}_a \tag{A.29}$$

Substituting the definition of $\underline{b}_a$ from equation 3.4 yields

$$\dot{\hat{\underline{b}}}_a + \delta\dot{\underline{b}}_a = -\frac{1}{\tau_a}\left(\hat{\underline{b}}_a + \delta\underline{b}_a\right) + \underline{n}_a \tag{A.30}$$

Distributing the $-\frac{1}{\tau_a}$ yields

$$\dot{\hat{\underline{b}}}_a + \delta\dot{\underline{b}}_a = -\frac{1}{\tau_a}\,\hat{\underline{b}}_a - \frac{1}{\tau_a}\,\delta\underline{b}_a + \underline{n}_a \tag{A.31}$$

Subtracting the definition of $\dot{\hat{\underline{b}}}_a$ found in Equation 3.9 yields the final form

$$\delta\dot{\underline{b}}_a = -\frac{1}{\tau_a}\delta\underline{b}_a + \underline{n}_a \tag{A.32}$$

The same process can be followed to obtain

$$\delta\dot{\underline{b}}_g = -\frac{1}{\tau_a}\delta\underline{b}_g + \underline{n}_g \tag{A.33}$$

Because the camera-to-body quaternion derivative $\dot{q}_c^b$ is a 0 vector, its linearization is simply a 0 vector as well.

APPENDIX B

Linearization of Measurements

Let the x, y, and z components of ℓ^c be ℓ_x, ℓ_y, and ℓ_z respectively, and let

$$h\left(\underline{\ell}^c\right) \equiv \begin{bmatrix} \frac{\ell_x}{\ell_z} \\ \frac{\ell_y}{\ell_z} \end{bmatrix} \tag{B.1}$$

Let the following be true:

$$\tilde{\underline{z}} \equiv \hat{\underline{\tilde{z}}} + \delta\underline{z} \tag{B.2}$$

$$\underline{\ell}^c \equiv \hat{\underline{\ell}}^c + \delta\underline{l}^c \tag{B.3}$$

$$\hat{\underline{\tilde{z}}} \equiv h\left(\hat{\underline{\ell}}^c\right) \tag{B.4}$$

For LOS measurements,

$$\tilde{\underline{z}} = h\left(\underline{\ell}^c\right) + \underline{\nu} \tag{B.5}$$

Substituting equations 3.14 and B.3 into Equation B.5 yields

$$\hat{\underline{\tilde{z}}} + \delta\underline{z} = h\left(\hat{\underline{\ell}}^c + \delta\underline{l}^c\right) + \underline{\nu} \tag{B.6}$$

Expansion using the Taylor series and dropping second-order terms yields

$$\hat{\underline{\tilde{z}}} + \delta\underline{z} = h\left(\hat{\underline{\ell}}^c\right) + \left(\frac{\partial h}{\partial \underline{\ell}^c}|_{\underline{\ell}^c = \hat{\underline{\ell}}^c}\right)\delta\underline{l}^c + \underline{\nu} \tag{B.7}$$

Subtraction of the nominal as defined in Equation B.4 yields

$$\delta \underline{z} = \left(\frac{\partial h}{\partial \underline{l}^b} |_{\underline{\ell}^c = \hat{\underline{\ell}}^c} \right) \delta \underline{\ell}^c + \underline{\nu} \tag{B.8}$$

Computing the Jacobian portion yields the H matrix H_ℓ:

$$H_\ell = \frac{\partial h}{\partial \underline{l}^b} |_{\underline{\ell}^c = \hat{\underline{\ell}}^c} = \begin{bmatrix} \frac{1}{\ell_z} & 0 & \frac{-\ell_x}{\ell_z^2} \\ 0 & \frac{1}{\ell_z} & \frac{-\ell_y}{\ell_z^2} \end{bmatrix} |_{\underline{\ell}^c = \hat{\underline{\ell}}^c} = \begin{bmatrix} \frac{1}{\hat{\ell}_z} & 0 & \frac{-\hat{\ell}_x}{\hat{\ell}_z^2} \\ 0 & \frac{1}{\hat{\ell}_z} & \frac{-\hat{\ell}_y}{\hat{\ell}_z^2} \end{bmatrix} \tag{B.9}$$

Equation B.8 is now in the form

$$\delta \underline{z} = H_\ell \delta \underline{l}^c + \underline{\nu} \tag{B.10}$$

Perturbations are then used to derive $\delta \underline{\ell}^c$.

Let $R_c^b \equiv \left[I - \left(\underline{\delta \theta}_c^b \times \right) \right] \hat{R}_c^b$, and transpose both sides to get $R_b^c \equiv \hat{R}_b^c \left[I + \left(\underline{\theta}_c^b \times \right) \right]$.

Also let $R_b^n \equiv \left[I - \left(\delta \underline{\theta}_b^n \times \right) \right] \hat{R}_b^n$, and transpose both sides to get

$$R_n^b \equiv \hat{R}_n^b \left[I + \left(\underline{\theta}_b^n \times \right) \right] \tag{B.11}$$

Also let $\underline{p}^n \equiv \hat{\underline{p}}^n + \delta \underline{p}^n$ as seen in Equation 3.4.

Referring to Equation 3.7, let $\underline{\ell}^c \equiv R_b^c R_n^b \left(\underline{r}_{f_i}^n - \underline{p}^n \right) - R_b^c \underline{d}^b$,

and therefore $\hat{\underline{\ell}}^c \equiv \hat{R}_b^c \hat{R}_n^b \left(\underline{r}_{f_i}^n - \hat{\underline{p}}^n \right) - \hat{R}_b^c \underline{d}^b$.

Using these equations and substituting in perturbations yields

$$\hat{\underline{\ell}}^c + \delta \underline{\ell}^c = \hat{R}_b^c \left[I + \left(\delta \underline{\theta}_c^b \times \right) \right] \left\{ \hat{R}_n^b \left[I + \left(\delta \underline{\theta}_b^n \times \right) \right] \left[\underline{r}_{f_i}^n - \hat{\underline{p}}^n - \delta \underline{p}^n \right] - \underline{d}^b \right\} \tag{B.12}$$

Distribution yields

$$\hat{\underline{\ell}}^c + \delta\underline{\ell}^c = \hat{R}_b^c \left[I + \left(\delta\underline{\theta}_c^b\times\right)\right] \left\{\hat{R}_n^b \left[\underline{r}_{f_i}^n - \hat{\underline{p}}^n - \delta\underline{p}^n + \left(\delta\underline{\theta}_b^n\times\right)\underline{r}_{f_i}^n - \left(\delta\underline{\theta}_b^n\times\right)\hat{\underline{p}}^n - \left(\delta\underline{\theta}_b^n\times\right)\delta\underline{p}^n\right] - \underline{d}^b\right\} \tag{B.13}$$

Further distribution, application of the anti-commutative property of the cross operator, and canceling second-order terms yields

$$\begin{aligned}\hat{\underline{\ell}}^c + \delta\underline{\ell}^c = \hat{R}_b^c \left\{\hat{R}_n^b \left[\underline{r}_{f_i}^n - \hat{\underline{p}}^n - \delta\underline{p}^n + \left(\hat{\underline{p}}^n - \underline{r}_{f_i}^n\right)\times\delta\underline{\theta}_b^n\right] - \underline{d}^b\right\} - \\ \hat{R}_b^c \left\{\left[\hat{R}_n^b \left(\underline{r}_{f_i}^n - \hat{\underline{p}}^n\right) - \underline{d}^b\right]\times\delta\underline{\theta}_c^b\right\}\end{aligned} \tag{B.14}$$

Subtracting the nominal and distribution yields

$$\delta\underline{\ell}^c = -\hat{R}_b^c\hat{R}_n^b\delta\underline{p}^n + \hat{R}_b^c\hat{R}_n^b\left(\hat{\underline{p}}^n - \underline{r}_{f_i}^n\right)\times\delta\underline{\theta}_b^n - \hat{R}_b^c\left[\hat{R}_n^b\left(\underline{r}_{f_i}^n - \hat{\underline{p}}^n\right) - \underline{d}^b\right]\times\delta\underline{\theta}_c^b \tag{B.15}$$

From this, another H matrix can be extracted

$$H_{los} = \begin{bmatrix} -\hat{R}_b^c R_n^b & 0_{3\times3} & \hat{R}_b^c\hat{R}_n^b\left[\left(\hat{\underline{p}}^n - \underline{r}_{f_i}^n\right)\times\right] & 0_{3\times3} & 0_{3\times3} & -\hat{R}_b^c\left\{\left[\hat{R}_n^b\left(\underline{r}_{f_i}^n - \hat{\underline{p}}^n\right) - \underline{d}^b\right]\times\right\}\end{bmatrix} \tag{B.16}$$

To obtain the overall form of $\delta\underline{z} = H\delta\underline{x} + \underline{\nu}$, the two H matrices can be multiplied together to yield the overall H matrix

$$H = H_\ell H_{los} \tag{B.17}$$

The combined H matrix can be separated into individual matrices to be multiplied by individual perturbations as shown in Equation B.18.

$$\delta \underline{\ell}^c = H_{\delta p^n} \delta \underline{p}^n + H_{\delta \theta_b^n} \delta \underline{\theta}_b^n + H_{\delta \theta_c^b} \delta \underline{\theta}_c^b + G \underline{\nu}_\ell \tag{B.18}$$

where

$$H_{\delta p^n} = -\hat{R}_b^c \hat{R}_n^b \tag{B.19}$$

$$H_{\delta \theta_b^n} = \hat{R}_b^c \hat{R}_n^b \left[\left(\underline{\hat{p}}^n - \underline{r}_{f_i}^n \right) \times \right] \tag{B.20}$$

$$H_{\delta \theta_c^b} = -\hat{R}_b^c \left\{ \left[\hat{R}_n^b \left(\underline{r}_{f_i}^n - \underline{\hat{p}}^n \right) - \underline{d}^b \right] \times \right\} \tag{B.21}$$

$$G = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{B.22}$$