8-2019

# Multiple IMU Sensor Fusion for SUAS Navigation and Photogrammetry

Matthew Givens
*Utah State University*

# MULTIPLE IMU SENSOR FUSION FOR SUAS NAVIGATION AND PHOTOGRAMMETRY

by

Matthew Givens

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

of

MASTER OF SCIENCE

in

Aerospace Engineering

Approved:

| | |
|---|---|
| David K. Geller, Ph.D.<br>Major Professor | Calvin R. Coopmans, Ph.D.<br>Committee Member |
| Stephen A. Whitmore, Ph.D.<br>Committee Member | Randall Christensen, Ph.D.<br>Committee Member |

Richard S. Inouye, Ph.D.
Vice Provost for Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2019

# ABSTRACT

Multiple IMU Sensor Fusion for sUAS Navigation and Photogrammetry

by

Matthew Givens, Master of Science

Utah State University, 2019

Major Professor: David K. Geller, Ph.D.
Department: Mechanical and Aerospace Engineering

The dominant paradigm in the implementation of inertial navigation systems in unmanned aerial systems remains that of using a single, high-precision inertial measurement unit (IMU) for attitude determination and control. Using an array of low-cost, MEMS IMUs in tandem could potentially offer a range of benefits above and beyond what a traditional configuration can deliver. Though the idea of fusing the outputs of multiple IMUs together has been examined in many prior works over the last 60 years, the approach of fusion in the estimation domain, wherein the outputs of multiple, independent local inertial navigation systems (INS) are fused inside a Kalman filter architecture with a single source of reference information such as GPS, has been relatively underexplored. In this thesis, both branches of IMU fusion will be explored, and a novel estimation domain strategy appropriate for implementation on a small unmanned aerial system will be derived and implemented in simulation.

(99 pages)

# PUBLIC ABSTRACT

Multiple IMU Sensor Fusion for sUAS Navigation and Photogrammetry

Matthew Givens

Inertial measurement units (IMUs) are devices that sense accelerations and angular rates in 3D so that vehicles and other devices can estimate their orientations, positions, and velocities. While traditionally large, heavy, and costly, using mechanical gyroscopes and stabilized platforms, the recent development of micro-electromechanical sensor (MEMS) IMUs that are small, light, and inexpensive has led to their adoption in many everyday systems such as cell phones, video game controllers, and commercial drones. MEMS IMUs, despite their advantages, have major drawbacks when it comes to accuracy and reliability. The idea of using more than one of these sensors in an array, instead of using only one, and fusing their outputs to generate an improved solution is explored in this thesis.

# ACKNOWLEDGMENTS

My path to this work, and indeed my path into engineering as a career, has been circuitous and often filled with uncertainty. Navigating the course and overcoming these and other challenges would not have been possible without the support of my family, my fiance, and the wonderful professors at Utah State University. I would especially like to thank my functional advisor, Dr. Calvin Coopmans, for giving me the opportunity to work at AggieAir and on this project, and Dr. Mac McKee from the Utah Water Research Laboratory for supporting the lab and this work.

Matthew Givens

CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# ACRONYMS

| | |
|---|---|
| sUAS | **s**mall **U**nmanned **A**erial **S**ystem |
| INS | **I**nertial **N**avigation **S**ystem |
| GNSS | **G**lobal **P**ositioning **S**atellite **S**ystem |
| GPS | **G**lobal **P**ositioning **S**ystem |
| IMU | **I**nertial **M**easurement **U**nit |
| MEMS | **M**icro-**E**lectro**m**echanical **S**ensor |
| EKF | **E**xtended **K**alman **F**ilter |
| AGL | **A**bove **G**round **L**evel |
| COTS | **C**ommercial **O**ff-**T**he-**S**helf |
| ECRV | **E**xponentially **C**orrelated **R**andom **V**ariable |
| OCFA | **O**ptimal **C**onstrained **F**usion **A**lgorithm |

# CHAPTER 1

# INTRODUCTION

The goal of this work is to explore ways of implementing multiple low-cost inertial measurement units (IMUs) on a small unmanned aerial system (sUAS). The necessary background and motivation for this research is presented in this chapter in order to contextualize the theoretical developments and simulation results in the following chapters.

## 1.1 Background on Inertial Navigation

A critical component in any navigation system is state estimation: the problem of determining a vehicle's orientation and position and predicting how those properties will change in the future. "Dead reckoning" is the idea of obtaining estimates of states like position, velocity, and attitude (PVA) by extrapolating information known from previous measurements of said states or related quantities. The inertial measurement unit (IMU) provides such measurements at high sampling rates and can be used for precise estimation of relative PVA states over short time intervals. IMUs typically implement some combination of rate gyroscopes and accelerometers to generate a raw output of angular rates and specific force in three dimensions. By integrating these quantities over time, an object's or vehicle's PVA states, relative to some starting state, can be estimated.

With perfect measurements, a dead reckoning strategy would be all that is required for precise navigation between known points in space. However, in practice, these measurements are far from perfect and will result in erroneous estimates and divergence if left uncorrected. Error sources include biases, scale factors, and misalignment errors that can be influenced by both manufacturing errors and external conditions such as temperature and radiation. Additionally, there is always random noise in the signal which cannot be predicted and must be compensated for. Instrument saturations and recommended operating ranges must also to be considered.

Because of the existence of measurement errors, pure dead reckoning is infeasible over any significant time interval for even the best IMUs. For this reason, IMUs are typically paired with other instruments, such as magnetometers and Global Navigation Satellite System (GNSS) receivers, in order to periodically bound estimates and sometimes enable estimation and compensation for some inertial measurement errors. A common algorithm for doing this is the Kalman filter, the optimal linear estimator. [3]

Using a statistical estimation scheme such as the Kalman filter allows information from any kind of sensor to be included in the navigation system. For instance, when an IMU is paired with a triaxial magnetometer, it is possible generate a complete estimate of the orientation of the system, including the compass or yaw angle. The estimate of an absolute yaw angle is otherwise unbounded to an unaided IMU due to that quantity's ambiguity with respect to the gravity vector. [1] In addition, since the magnetometers are measuring an (ideally) inertially fixed phenomenon, (the Earth's magnetic field) the errors of the other measurements of the IMU can be bounded. This type of system is known as an Attitude and Heading Reference System (AHRS) and is common in applications that need simple, stable attitude solutions. The AHRS does not provide information about absolute position or velocity but can be used for short periods for improved dead reckoning.

Another sensor commonly integrated with an IMU is the GNSS receiver, of which the most common is are Global Positioning System (GPS) receivers. GNSS systems are constellations of satellites in precisely-known orbits about the Earth that produce signals that can be picked up by receivers. A civilian device's GPS receiver periodically solves a system of equations and determines its absolute position on the surface of the Earth to within, at best, approximately $\pm 1$ meter in the horizontal directions and $\pm 3$ meters in the vertical direction. However, the frequency at which GPS signals can be processed is relatively low, around 1 Hz. A system that combines these low-rate GPS measurements with higher-rate IMU measurements to estimate PVA, perhaps with the aid of other instruments, is called an Inertial Navigation System (INS).

## 1.2 The Inertial Measurement Unit

Traditional IMUs, typically used for ships, aircraft, spacecraft, and missiles, used gimballed, actively stabilized platforms in order to keep the individual mechanical sensors inside the IMU level with an "inertial" reference frame such as a local North-East-Down (NED) or Earth-Centered-Earth-Fixed (ECEF) frames. These systems can achieve great performance but suffer from disadvantages of high complexity, high cost, and possible susceptibility to mechanical gimbal lock. More recently, a "strapdown" configuration has become more common in which the sensors are rigidly mounted to the vehicle and extra computation is done onboard in order to record its orientation relative to the inertial coordinate frame.

The quality and cost of IMU types spans orders of magnitude, from the lowest quality sensors being used in destructive automotive crash testing to the highest, most costly being used in large ships and spacecraft. A collection of magnitudes of typical errors and cost of each level of sensor is included in Table 1.1 below.

Table 1.1: Typical Accelerometer and Gyro Biases for Different Grades of IMU [1] [2]

| *IMU Grade* | *Accelerometer Bias (mg)* | *Gyro Bias (deg/hr)* | *Cost (USD $)* |
|---|---|---|---|
| Marine | 0.01 | 0.001 | $\geq$1,000,000 |
| Aviation | $0.03 - 0.1$ | 0.01 | 100,000 |
| Intermediate | $0.1 - 1$ | 0.1 | 10,000 |
| Tactical | $1 - 10$ | $1 - 100$ | 1,000 |
| Automotive | $> 10$ | $> 100$ | 10-100 |

In the last few decades, micro-electromechanical sensors (MEMS) that can fit on small printed circuit boards have emerged and offer the benefits of vastly reduced weight, size, and power consumption. While this inevitably comes at the cost of performance, the advent of strapdown MEMS IMUs has led to their application in hundreds of new products such as micro-sized drones, cell phones, and smart watches to name a few. MEMS IMUs typically fall under the "automotive" grade, the lowest-cost, highest uncertainty category.

### 1.3 Research Motivation

AggieAir [4], a research laboratory at Utah State University specializing in designing, building, and flying small Unmanned Aerial Systems (sUAS) for the purposes of scientific data collection, flies its sUAS at altitudes ranging from 400 feet AGL to 3000 feet AGL. The latter is considered high altitude flight for non-military sUAS at the time of this writing and presents the added challenge of acquiring sufficiently precise pointing data for the high-precision camera payload to meet its requirements.

MEMS IMUs can produce inertial information accurate enough for adequate navigation and control of most sUAS when paired with a GNSS system. However, in the context of aerial imaging for scientific data collection, wherein every image must be georeferenced and stitched together into a mosaic, typical low-cost sensors are generally inadequate. Furthermore, as an imaging aircraft increases its altitude above ground level (AGL), the pointing requirements for its cameras become ever more stringent. Figure 1.1 illustrates this concept.

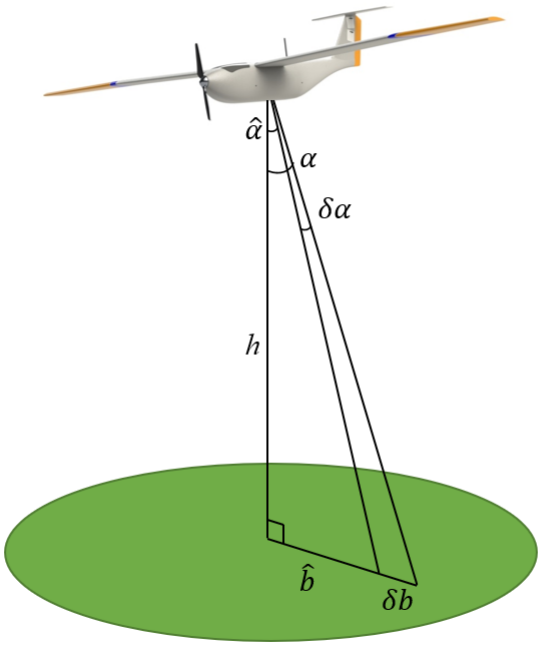


Fig. 1.1: As $h$ increases, the effect that errors in the pointing angle $\delta\alpha$ have on the spatial resolution $\delta b$ increases linearly (See Appendix A for derivation)

Currently, AggieAir implements a MEMS-based navigation solution in its aircraft called the VectorNav VN-200. This sensor package was chosen after analysis and deliberation outlined in [5] and costs in excess of $2,400 USD for a single calibrated unit. The VN-200 and related products use a low-cost MEMS IMU but add sophisticated filtering, onboard GPS processing, and rigorous calibration to the sensor package in order to meet an advertised pitch and roll attitude uncertainty specification of $\pm 0.1$ degrees and heading uncertainty specification of $\pm 0.3$ degrees.

While the VN-200 has proven to be a reliable navigation and control solution, its relatively high cost in the context of sUAS has motivated a search for lower-cost alternatives.

### 1.4 Multiple IMUs

As computer technology continues to advance at a rapid pace, more computationally-intensive multisensor architectures have become more feasible. One such idea is that of fusing the outputs of multiple IMUs into a combined solution with the possible benefits of better performance, fault detection and isolation (FDI), direct error estimation, and more. Such an algorithm could potentially be used to fuse the outputs of low-cost MEMS sensors into a package that would have sufficient performance for flight onboard an AggieAir-style sUAS. Further, the same algorithm could be used to fuse multiple VN-200s or higher-cost sensors for even greater fidelity. The increased redundancy and resulting robustness would also strengthen AggieAir's safety and reliability for Beyond Visual Line of Sight (BVLOS) missions. This paradigm sets the backdrop for this thesis.

There has been significant prior work involving multiple accelerometer, multiple gyroscope, and multiple IMU sensor constellations such that a full literature review is necessary. Much of the work in the following chapter was published by the author in [6] with many more sources and is summarized here.

### 1.5 Contribution

With the goal of designing and modeling a multi-IMU system with improved performance and redundancy for application in sUAS navigation and photogrammetry, this thesis

shows the results of two state-of-the-art observation-domain methods in Chapter 4 and presents the derivation and implementation of a novel solution to the estimation domain centralized filter strategy in Chapter 5. In order to accomplish this, an Extended Kalman Filter (EKF) framework based on an aircraft simulation is described in Chapter 3 and implemented in MATLAB. A geometry study is conducted in Chapter 6 in order to characterize the behavior of the proposed fusion strategy. In addition to this application, it is anticipated that this project will lay the groundwork for future multi-sensor fusion projects at Utah State University.

# CHAPTER 2

# LITERATURE REVIEW

The idea of using multiple IMUs has been around for many years and has been implemented in a variety of ways. The following sections outline some basic distinctions in the way this idea has been approached in the past with emphasis on sUAS applications.

## 2.1 Categorization

A useful distinction made in [7] and [8] that will be adopted here is the idea of separating multiple IMU fusion approaches into two categories: observation domain and estimation domain. The observation domain category encompasses strategies that attempt to combine data into a virtual IMU (VIMU) sensor with a defined sensor frame before being inputted into a INS/GNSS Kalman filter or similar architecture for PVA state estimation. The filter itself therefore only "sees" a single VIMU and conventional navigation filtering strategies can be employed without alteration. This is the most natural, and therefore most thoroughly researched [7], approach to the MIMU problem. The disadvantage of this approach is that it can be difficult to account for the individual sensor error characteristics and biases, particularly when the sensors have varying specifications. Precise timing is also a requirement. Many of the works in this category do not extend into INS/GNSS state estimation architectures because they are focused on the fusion of data in the observation domain to produce improved estimates of angular rates and specific force. A diagram depicting the observation domain fusion strategy is provided in Figure 2.1.

Estimation domain strategies, also referred to as fusion in the state space domain by some authors [9], seek to combine individual PVA state estimates derived from multiple local INS filters into a combined state estimate, either after every "propagation step" or "update step." While generally more computationally intensive, this idea has the advantage of fusing individually calculated, error-corrected state estimates and thus offers easier modularity

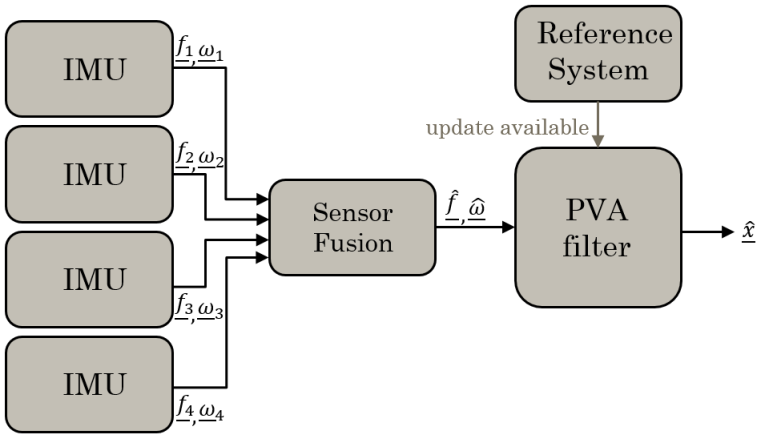


Fig. 2.1: Example of observation domain Sensor Fusion

for IMU sensor combinations and increased fault detection capability. There are fewer sources that use this strategy explicitly for INS/GNSS navigation in the literature than for observation domain fusion, although a deep body of literature exists on this topic in the more generic context of multi-sensor fusion. An example of an estimation domain architecture is shown in Figure 2.2.

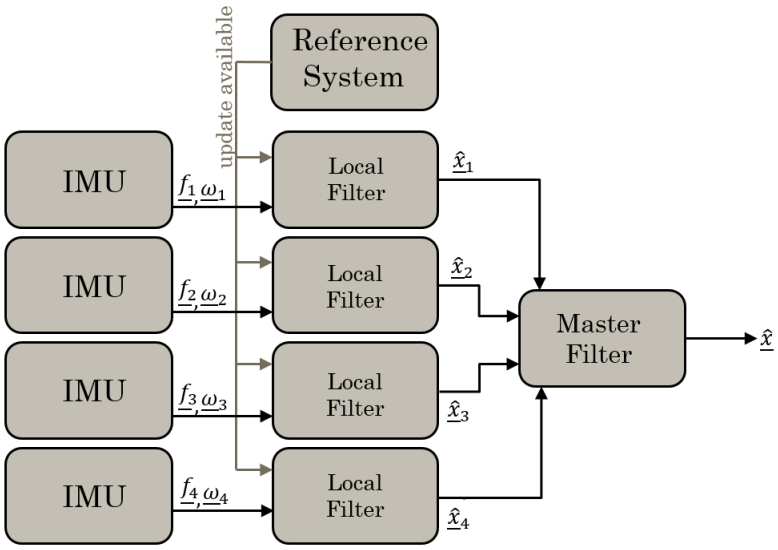


Fig. 2.2: Example of Estimation Domain Sensor Fusion

## 2.2 Observation domain Estimation

### 2.2.1 Gyro-Free Methods

A large part of the literature is dedicated to gyro-free IMU (GF-IMU), also known as all-accelerometer arrays, as a basis for redundant IMU fusion techniques. Many of the same results for arbitrary, singe-axis linear accelerometer arrays can be applied to all-accelerometer arrays that implement accelerometer triads [10].

The first works on inertial sensor fusion took place during rapid development of space navigation science and technology in the 1960s when idea of using individual, geometrically separated linear accelerometers that span 3D space to estimate angular states was first proposed and studied [11] [12]. These concepts were an attractive topic then and in the following decades due to the often prohibitive cost and complexity of gyroscopes as compared to accelerometers at the time. The idea behind a GF-IMU comes from an intuitive application of classic rigid-body dynamics. The Coriolis formula relates the acceleration at an arbitrary discrete point on a rigid body, in this case a vehicle, to the kinematics of the rigid body as a whole:

$$\underline{f}_i = \underline{f} + \underline{\omega} \times (\underline{\omega} \times \ \underline{r}_i) + \dot{\underline{\omega}} \times \underline{r}_i \tag{2.1}$$

where $\underline{f}_i$ is the specific force at an $i$-th arbitrary point, $\underline{\omega}$ is the angular velocity of the rigid body, and $\underline{r}_i$ is the distance from the origin or center-of-mass to the $i$-th point in question, which will hereafter be referred to as the "lever-arm." The specific force at the origin of the vehicle frame is defined as

$$\underline{f} = \underline{a} - \underline{g} \tag{2.2}$$

where $\underline{a}$ is the total acceleration and $\underline{g}$ is the known acceleration due to gravity. From this formulation, it can be intuited that if the accelerations at discrete points on the vehicle are measured and their locations are known, it may be possible to estimate the rigid body's overall linear acceleration, its angular acceleration, and perhaps even its angular velocity given enough unique accelerometer measurements.

There are many sources, starting in the 1960s and going through today, that discuss

and analyze the all-accelerometer approach. Many are provided in the bibliography of this work but the reader is referred to a literature survey by Nilsson et al. [13] for an exhaustive list of sources as well as a summary of the key developments of the field through the years.

With the rise of MEMS technology, there has been a general shift in all-accelerometer IMU research away from individual single-axis accelerometers to the more commonly available triaxial ones. Given this context, [14] provides a very useful least-squares formulation for the direct calculation of angular acceleration and velocity using a minimum of four accelerometer triads that span 3D space. The method is summarized in [8] and repeated in the Chapter 4 of this thesis. A similar formulation was used by [15] in 2005 to determine the feasibility of all-accelerometer concept on sUAS and concluded that it was inadequate with the sensors available at the time.

With the rise of very small, inexpensive MEMS gyroscopes in the early 2000s, the gyro-free IMU concept saw a decrease in popularity. However, [16] revisited the study of all-accelerometer architectures with a comprehensive derivation of GF-IMU theory and mechanization in the context of cold-atom interferometry-based, high-precision accelerometers that have the potential to vastly outperform older technology in GF-IMU applications.

### 2.2.2 Redundant IMUs

While all-accelerometer arrays leverage distributed geometry in order estimate angular velocity, redundant IMUs (RIMUs) have now generally come to include rate gyroscopes, removing the necessity of non-negligible lever-arms between sensors. This allows for RIMUs to be built or sold as a complete sensor package and oftentimes on a single printed circuit board as in [8] and [17]. An important result, originating in [18] (from 1974), is that an optimal configuration of redundant IMUs in terms of fault tolerance is a "skewed" configuration, wherein the sensitive axes of each triaxial IMU sensor are not aligned and transformations are carried out to align them in a virtual frame. This is known as a "skewed-redundant" IMU (SRIMU) and has been examined in many sources such as [19] [20] [21], and is shown in Fig. 2.3. In the context of MEMS sensors, [22] showed that navigation performance improvements of up to 50% can be obtained using a skewed-redundant configuration of

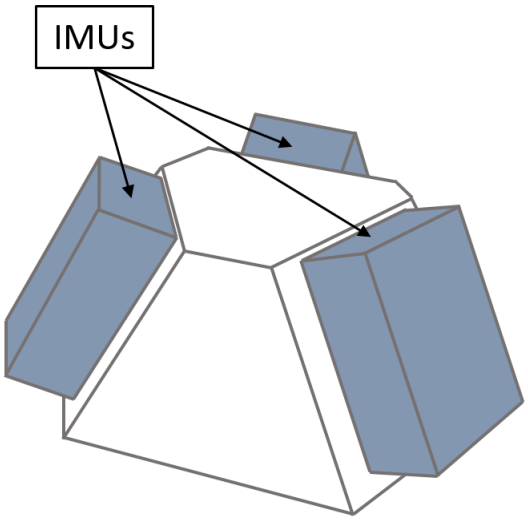


Fig. 2.3: Example of a skewed-redundant IMU configuration

MEMS IMUs. Guerrier [23], however, proves that the optimality of IMU sensor triads is actually independent of the geometry between them as long as there are no sensor failures. Further, [23] and [21] state that the advantage of skewed-redundant configurations over orthogonally redundant configurations is only preferable when high-cost sensors are involved because the advantages of skewed-redundant configurations are reduced when larger numbers of IMUs are implemented.

A form of the least squares approach mentioned in the previous section is used by many authors in the context of redundant IMUs along with other fusion techniques that take advantage of the available gyroscope data such as [24] and [25]. [8] contrasts the least-squares approach with a novel maximum likelihood estimation scheme that outperforms it, relaxes the requirement of having a three dimensional inertial array, and can be readily combined with redundant gyroscope information. This algorithm will be stated and implemented in Section 5.2. Recently, the same research group has further extended that work in [26] with the addition of a motion model.

Redundant IMUs have been implemented on sUAS only a few times. Sukkarieh [27] constructed a skewed-redundant inertial IMU using larger IMUs in an information filter. Recently, [28] used an optimal filter based on the Wiener method to improve the accuracy

of position determination.

## 2.3 Estimation Domain Methods

There exists a wide field of research dedicated to the general problem of multi-sensor fusion, and this research has produced algorithms, generally based on varieties of the Kalman or information filters, that can be used in the standard INS/GNSS sensor fusion problem. It should be noted that many of these methods are applied to the observation domain sensor fusion problem to estimate specific force or angular velocity states. The distinction meant by "estimation domain" here is the idea of fusing the solutions to multiple INS, generally PVA states, in local filters into a combined one via a "master filter".

The major advantage of estimation domain architectures is that statistics and error characteristics of each sensor can be individually accounted for because each estimate has already been corrected for biases and other errors in each local filter. The implementation of voting algorithms for fault detection and isolation is also made straightforward. The first application of an estimation domain technique to multiple INS appears to be [9] where relative orientation constraints were used in the observation model. Bancroft [7] delineates two possible estimation domain strategies: centralized and federated, and most authors seem to use a federated approach.

### 2.3.1 Centralized Filters

The centralized architecture generally consists of a "block" or "stacked" filter that computes individual uncorrelated state estimates in a single block diagonal structure:

$$\delta \underline{x}_{k+1} = \mathbf{\Phi}_{k,k+1} \delta \underline{x}_k + \underline{w}_k \tag{2.3}$$

$$\delta \underline{z}_{k+1} = \mathbf{H}_{k,k+1} \delta \underline{x}_k + \underline{\eta}_k \tag{2.4}$$

where

$$\delta \underline{x}_{k+1} = \begin{bmatrix} \delta \underline{x}_{1,k+1} \\ \delta \underline{x}_{2,k+1} \\ \vdots \\ \delta \underline{x}_{n,k+1} \end{bmatrix} \qquad \delta \underline{x}_{k} = \begin{bmatrix} \delta \underline{x}_{1,k} \\ \delta \underline{x}_{2,k} \\ \vdots \\ \delta \underline{x}_{n,k} \end{bmatrix} \tag{2.5}$$

$$\underline{w}_{k} = \begin{bmatrix} \underline{w}_{1,k} \\ \underline{w}_{2,k} \\ \vdots \\ \underline{w}_{n,k} \end{bmatrix} \qquad \underline{\eta}_{k} = \begin{bmatrix} \underline{\eta}_{1,k} \\ \underline{\eta}_{2,k} \\ \vdots \\ \underline{\eta}_{n,k} \end{bmatrix} \tag{2.6}$$

$$\mathbf{\Phi}_{k,k+1} = \begin{bmatrix} \mathbf{\Phi}_{1,k,k+1} & 0 & \dots & 0 \\ 0 & \mathbf{\Phi}_{2,k,k+1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{\Phi}_{n,k,k+1} \end{bmatrix} \tag{2.7}$$

$$\mathbf{H}_{k+1} = \begin{bmatrix} \mathbf{H}_{1,k+1} & 0 & \dots & 0 \\ 0 & \mathbf{H}_{2,k+1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{H}_{n,k+1} \end{bmatrix} \tag{2.8}$$

in which $\delta \underline{x}_{ik}$ represents $i$-th the error state vector, $\delta \underline{z}_i^k$ are the misclosure vectors, $\mathbf{\Phi}_{k,k+1}$ are the state transition matrices, $\mathbf{H}_{k+1}$ are the measurement geometry vectors, and $\delta \underline{w}_k$ and $\eta_k$ are the zero-mean, white Gaussian process and measurement noises respectively.

As detailed in [29], the centralized architecture is seemingly not viable when there is only one source of GPS measurements because time correlations are introduced into the observations. If previous knowledge of the time correlation characteristics is known, [30] shows that it is possible to mitigate this effect. It is also possible to use relative position updates using known geometry constraints, as in [7] and [9], in order to bound the estimates. Overall, the centralized filter approach remains relatively understudied in this context. A visual representation of a centralized block filter is shown in Fig. 2.4.

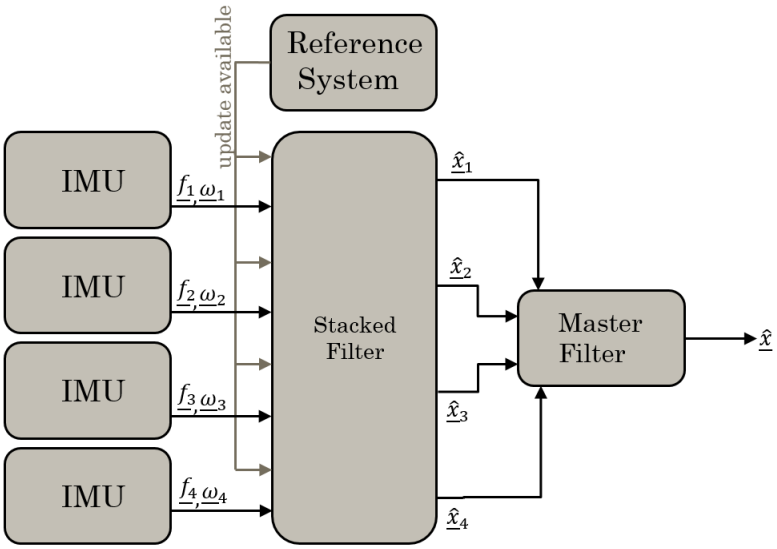


Fig. 2.4: Example of the "stacked" or "block" filter estimation domain strategy

### 2.3.2 Federated Filters

In 1988, Carlson [31] outlined the mathematical framework for a general multisensor federated fusion architecture that has been used in multiple works since including [17] and [32]. The federated filter uses a Kalman filter or other master fusion algorithm to fuse the outputs of independent sensors into one estimate. In contrast to the stacked filter, these measurements are assumed to be completely separate and are not computed together before fusion. Carlson's 2002 paper [33] is an authoritative resource on the federated filter in a general context.

The "Federated No Reset" (FNR) approach is one such federated algorithm, as described in [1] and [29]. The state estimates are combined using a "snapshot" fusion method that contains no *a priori* information and assumes that the local filter estimates are independent. To accomplish the snapshot filtering optimally, the following information conservation approach, as derived in [33], can be used:

$$\mathbf{P}_{eq}^{-1} = \sum_{i=1}^{n} \mathbf{P}_{i}^{-1} \tag{2.9}$$

$$\underline{x}_{eq} = \mathbf{P}_{eq}^{-1} \sum_{i=1}^{n} \mathbf{P}_{i}^{-1} \underline{x}_{i} \tag{2.10}$$

where $\mathbf{P}_{eq}$ is the covariance of the combined estimate and $\underline{x}_{eq}$ is the combined state estimate itself. For a visual representation of an FNR architecture, the reader can refer back to Fig. 2.2. It should be noted that other, more complex federated filtering strategies exist and information about those can be found in [1] and [29].

## 2.4 Errata

The literature review published by the author [6] contains a significant error that will be addressed here. The "information conservation" approach was incorrectly labeled as a "least-squares" approach in that work. This mistake was caught after publication and may cause some confusion among readers, so it is hoped that they will find this correction.

# CHAPTER 3

# CENTRALIZED GNSS/INS EXTENDED KALMAN FILTER

This chapter develops the centralized Extended Kalman Filter (EKF) algorithm that is implemented and is used to analyze the IMU and INS fusion methods in the following chapters.

## 3.1 Coordinate Frames

A proper mathematical development of the algorithm used in this work requires the specification of variable notation and coordinate frames. Following a development shown in [1], three types of coordinate frames used are:

- Object frame: the frame attached to the object and whose motion is being described denoted below as $\alpha$
- Reference frame: the frame in which the motion of the object frame is expressed, denoted as $\beta$
- Resolving frame: the frame in which the motion is represented, denoted as $\gamma$

A kinematic property requires these frames to be specified, although sometimes they can seem redundant. Velocity is an example of a quantity that requires all three of these frames to be properly specified.

$$\underline{v}^{\beta}_{\alpha/\gamma} \tag{3.1}$$

represents the angular velocity of the object frame $\alpha$ with respect to the reference frame $\beta$ expressed in the resolving frame $\gamma$. To simplify the notation, the object and resolving frame specification will be dropped in the following sections. The resolving frame will be specified when necessary.

### 3.1.1 Body Frame

The aircraft in the following derivation has a "true" body frame that is used for reference. Each IMU is given its own body frame that is initially aligned with the true body frame of the vehicle. All of these body frames are orthonormal with the first axis pointing out of the nose of the aircraft, the second axis pointing out the starboard wing, and the third axis downward, completing the triad.

### 3.1.2 North-East-Down Frame

In order to specify the motion of the body frame(s), an "inertial" reference frame must be chosen. In this case, a North-East-Down (NED) frame was chosen in order to simplify some calculations. This frame is generally sufficiently inertial for sUAS because but becomes increasingly unrealistic for higher altitudes and longer distances because it assumes a locally flat, nonrotating Earth.

The NED frame is an orthonormal triad with the first axis pointing toward true north along a sphere and the third axis pointing down to the center of the sphere. The second axis completes a right-handed triad. The shape of the Earth and the location of the local NED frame were not considered in this work, but it can be extended to handle it.

## 3.2 Truth Model

A true trajectory was generated using a flight simulation code written by fellow USU graduate student Jackson Graham. This provided true body-frame acceleration and angular rate data at 100 Hz at the center of mass of the aircraft. A plot of this true trajectory is shown in Fig. 2.1.

Even though multiple IMUs were considered and will have individual state estimates, the sUAS itself, taken as a single rigid body, has a single truth state for position, velocity, and attitude. The true biases for each IMU are also part of the state vector and must be propagated with different values of white, Gaussian random noise for each Monte Carlo run. Because of this, the truth state vector's dimension will be $10 + 6n \times 1$ where $n$ is the

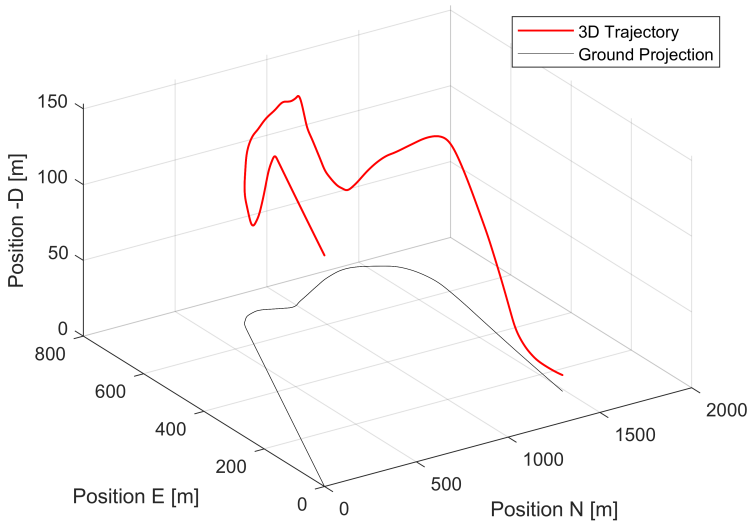


Fig. 3.1: True trajectory of aircraft from flight simulation with 2D ground track (black line)

number of IMUs. This truth state vector can be written as:

$$\underline{x}_{true} = \begin{bmatrix} \underline{r}_b^{ned} \\ \underline{v}_b^{ned} \\ q_{ned}^b \\ \underline{b}_{a,i}^b \\ \underline{b}_{g,i}^b \\ \underline{b}_{a,i+1}^b \\ \underline{b}_{g,i+1}^b \\ \vdots \end{bmatrix} \tag{3.2}$$

where $\underline{r}_b^{ned}$ and $\underline{v}_b^{ned}$ are the position and velocity of the vehicle body with respect to the NED frame expressed in the NED frame. $q_{ned}^b$ is a quaternion representing the attitude of the vehicle, $\underline{b}_{g,i}^b$ is the $i$-th vector of gyroscope biases in the body frame, and $\underline{b}_{a,i}^b$ is the $i$-th vector of accelerometer biases in the body frame. Note that the last element of the quaternion is taken to be the scalar part. The truth states are propagated forward using a fourth-order Runge-Kutta integrator because a different (unknown) integrator was used in

the flight simulation. The kinematics of the position, velocity, and attitude are described by

$$\dot{\underline{r}}^{ned} = \underline{v}^{ned} \tag{3.3}$$

$$\dot{\underline{v}}^{ned} = \mathbf{R}_b^{ned} \underline{f}^b + \underline{g}_{ned} \tag{3.4}$$

$$\dot{q}_{ned}^b = \frac{1}{2} \begin{bmatrix} \underline{\omega}^b \\ 0 \end{bmatrix} \otimes q_{ned}^b \tag{3.5}$$

where $\underline{g}_{ned}$ assumed to be constant. The true bias states are synthesized as first order Markov processes with white noise $\underline{w}$ and constants $\tau$.

$$\dot{\underline{b}}_{a,i} = -\frac{1}{\tau_{a,i}} \underline{b}_{a,i}^b + \underline{w}_a \tag{3.6}$$

$$\dot{\underline{b}}_{g,i} = -\frac{1}{\tau_{g,i}} \underline{b}_{g,i}^b + \underline{w}_g. \tag{3.7}$$

The true accelerometer measurements at the arbitrary locations of the individual IMUs, represented below as specific force $\underline{f}_i^b$, can be computed using the equation 2.1 along with the true accelerations and angular rates obtained from the flight simulation:

$$\underline{f}_i^b = \underline{f}^b + (\dot{\underline{\omega}}^b \times \underline{l}_i^b) + \underline{\omega}^b \times (\underline{\omega}^b \times \underline{l}_i^b) \tag{3.8}$$

where $\underline{l}_i^b$ is the position of the $i$-th sensor in the body frame.

### 3.3 Design Model

The design model encompasses the positions, velocities, and attitudes of each sensor in addition to their individual accelerometer and gyro bias vectors. The design state vector is

$$\underline{x}_{design} = \begin{bmatrix} \underline{r}_i^{ned} \\ \underline{v}_i^{ned} \\ q_{ned,i}^{b} \\ \underline{b}_{a,i}^{b} \\ \underline{b}_{g,i}^{b} \\ \underline{r}_{i+1}^{ned} \\ \underline{v}_{i+1}^{ned} \\ q_{ned,i+1}^{b} \\ \underline{b}_{a,i+1} \\ \underline{b}_{g,i+1} \\ \vdots \end{bmatrix} \tag{3.9}$$

The dynamics can be represented by

$$\dot{\underline{x}}_{design} = f(\underline{x}, \tilde{\underline{y}}, \underline{w}) = \begin{bmatrix} \dot{\underline{r}}_i^{ned} \\ \dot{\underline{v}}_i^{ned} \\ \dot{q}_{ned,i}^{b} \\ \dot{\underline{b}}_{a,i} \\ \dot{\underline{b}}_{g,i} \\ \vdots \end{bmatrix} \tag{3.10}$$

Since a mathematical model of the dynamics is very difficult to obtain in the case of atmospheric flight, *model replacement* is used here. The dynamic equations are transformed to kinematic equations based on measurements from the sensors in the IMUs.

$$\dot{\underline{r}}_i^{ned} = \underline{v}_i^{ned} \tag{3.11}$$

$$\dot{\underline{v}}_i^{ned} = \mathbf{R}_{b,i}^{ned}(\tilde{\underline{f}}_i^b - \underline{b}_{a,i}^b - \underline{n}_{f,i}) + \underline{g}_{ned,i} \tag{3.12}$$

$$\dot{q}_{ned,i}^b = \frac{1}{2}\begin{bmatrix} \tilde{\underline{\omega}}_i^b - \underline{b}_{g,i}^b - \underline{n}_{\omega,i} \\ 0 \end{bmatrix} \otimes q_{ned,i}^b \tag{3.13}$$

$$\dot{\underline{b}}_{a,i} = -\frac{1}{\tau_{a,i}}\underline{b}_{a,i}^b + \underline{w}_{a,i} \tag{3.14}$$

$$\dot{\underline{b}}_{g,i} = -\frac{1}{\tau_{g,i}}\underline{b}_{g,i} + \underline{w}_{g,i} \tag{3.15}$$

where $\tilde{\underline{f}}_i^b$ is the specific force measured by the accelerometers in the aircraft's body frame, calculated after integrating the truth state, $\tilde{\underline{\omega}}_i^b$ is the angular rates measured by the gyroscopes, the $\tau_i$ terms are time constants associated with the first order Markov process, and $\underline{w}$ terms represent process noise. The inertial measurement vector is

$$\tilde{\underline{y}}_i^b = \begin{bmatrix} \tilde{\underline{f}}_i^b \\ \tilde{\underline{\omega}}_i^b \end{bmatrix} = \begin{bmatrix} \underline{f}_i^b + \underline{b}_{a,i}^b + \underline{n}_{f,i} \\ \underline{\omega}_i^b + \underline{b}_{g,i}^b + \underline{n}_{\omega,i} \end{bmatrix} \tag{3.16}$$

A vector of process noise terms for each sensor can be written as:

$$\underline{w}_i = \begin{bmatrix} \underline{n}_{f,i} \\ \underline{n}_{\omega,i} \\ \underline{w}_{a,i} \\ \underline{w}_{g,i} \end{bmatrix} \tag{3.17}$$

where the power spectral density is defined as:

$$E\left[\underline{w}_i(t)\underline{w}'^T_i\right] = \mathbf{Q}_{w,i}\delta(t-t') \tag{3.18}$$

and

$$\mathbf{Q}_{w,i} = \text{diag}\left(\mathbf{Q}_{f,i}, \mathbf{Q}_{\omega,i}, \frac{2\sigma_{a,ss,i}^2}{\tau_a} I_{3\times 3}, \frac{2\sigma_{g,ss,i}^2}{\tau_g} I_{3\times 3}\right) \tag{3.19}$$

where $\sigma_{a,ss}^2$ and $\sigma_{g,ss}^2$ are the steady state variances of the accelerometer and gyroscope biases respectively.

### 3.4 State Propagation

Taking the expected value of the design model yields the navigation model:

$$\hat{\underline{\dot{r}}}_i^{ned} = \underline{\hat{v}}_i^{ned} \tag{3.20}$$

$$\hat{\underline{\dot{v}}}_i^{ned} = \mathbf{R}_{b,i}^{ned}(\underline{\tilde{f}}_i^b - \underline{\hat{b}}_{a,i}^b) + \underline{g}_{ned,i} \tag{3.21}$$

$$\hat{\dot{q}}_{ned,i}^b = \frac{1}{2}\begin{bmatrix} \underline{\tilde{\omega}}_i^b - \underline{\hat{b}}_{g,i}^b \\ 0 \end{bmatrix} \otimes \hat{q}_{ned,i}^b \tag{3.22}$$

$$\hat{\underline{\dot{b}}}_{g,i} = -\frac{1}{\tau_{a,i}}\underline{\hat{b}}_{a,i}^b \tag{3.23}$$

$$\hat{\underline{\dot{b}}}_{g,i} = -\frac{1}{\tau_{g,i}}\underline{\hat{b}}_{g,i}^b. \tag{3.24}$$

where $\otimes$ is a quaternion product. These equations are propagated forward at 100 Hz and naturally show divergence without additional measurement update information. This will be provided by the GPS measurements in section 3.6.

### 3.5 Covariance Propagation

The design model must be linearized in order to properly propagate the covariance. A perturbation approach can be used wherein each of the states are perturbed by the addition of a small error:

$$\underline{r}_i^{ned} = \underline{\hat{r}}_i^{ned} + \delta\underline{r}_i^{ned} \tag{3.25}$$

$$\underline{v}_i^{ned} = \underline{\hat{v}}_i^{ned} + \delta\underline{v}_i^{ned} \tag{3.26}$$

$$\underline{b}_{a,i}^b = \underline{\hat{b}}_{a,i}^b + \delta\underline{b}_{a,i}^b \tag{3.27}$$

$$\underline{b}_{g,i}^b = \underline{\hat{b}}_{g,i}^b + \delta\underline{b}_{g,i}^b \tag{3.28}$$

The perturbed attitude state, in the form of a quaternion, must be represented differently than the other states:

$$q_{ned}^b = \hat{q}_{ned}^b \otimes \delta q_{ned}^b \tag{3.29}$$

where the perturbation is a small rotation from the estimate:

$$\delta q^b_{ned} = \begin{bmatrix} \delta\underline{\theta}_b/2 \\ 1 \end{bmatrix} \tag{3.30}$$

Inserting these perturbation equations into the design model yields:

$$\dot{\hat{\underline{r}}}^{ned}_i + \delta\dot{\underline{r}}^{ned}_i = \hat{\underline{v}}^{ned}_i + \delta\underline{v}^{ned}_i \tag{3.31}$$

$$\dot{\hat{\underline{v}}}^{ned}_i + \delta\dot{\underline{v}}^{ned}_i = \mathbf{R}^{ned}_b(\tilde{\underline{f}}^b - \hat{\underline{b}}_{a,i} - \delta\underline{b}^b_{a,i} - \underline{n}_{f,i}) + \underline{g}_{ned,i} \tag{3.32}$$

$$\frac{d}{dt}(\delta q^b_{\hat{b}} \otimes q^{\hat{b}}_{ned}) = \delta\dot{q}^b_{\hat{b}} \otimes q^{\hat{b}}_{ned} + \delta q^b_{\hat{b}} \otimes \dot{q}^{\hat{b}}_{ned} \tag{3.33}$$

$$\dot{\hat{\underline{b}}}_{a,i} + \delta\dot{\underline{b}}_{a,i} = -\frac{1}{\tau_a}(\hat{\underline{b}}_{a,i} + \delta\underline{b}^b_{a,i}) + \underline{w}_{a,i} \tag{3.34}$$

$$\dot{\hat{\underline{b}}}_{g,i} + \delta\dot{\underline{b}}_{g,i} = -\frac{1}{\tau_g}(\hat{\underline{b}}_{g,i} + \delta\underline{b}^b_{g,i}) + \underline{w}_{g,i} \tag{3.35}$$

The attitude estimation error must be defined in terms of the direction cosine matrix:

$$\mathbf{R}^b_{ned} = \mathbf{R}^b_{\hat{b}}\mathbf{R}^{\hat{b}}_{ned} \tag{3.36}$$

Rearranging to solve for the DCM from the body frame to the NED frame:

$$\mathbf{R}^{ned}_b = \mathbf{R}^{ned}_{\hat{b}}(\mathbf{R}^b_{\hat{b}})^T \tag{3.37}$$

The small rotation $\mathbf{R}^b_{\hat{b}}$ can be represented as

$$\mathbf{R}^b_{\hat{b}} = [\mathbf{I} - (\delta\underline{\theta}_b\times)] \tag{3.38}$$

where $(\delta\underline{\theta}_b\times)$ is a skew-symmetric cross product matrix. The previous velocity derivative equation can therefore be represented as

$$\dot{\hat{\underline{v}}}^{ned}_i + \delta\dot{\underline{v}}^{ned}_i = \mathbf{R}^{ned}_{\hat{b},i}[\mathbf{I} - (\delta\underline{\theta}_{b,i}\times)](\tilde{\underline{f}}^b - \hat{\underline{b}}_{a,i} - \delta\underline{b}^b_{a,i} + \underline{n}_{f,i}) + \underline{g}_{ned,i} \tag{3.39}$$

The form that these equations must take in the linearization is:

$$\delta\dot{\underline{x}} = \hat{\mathbf{F}}\delta\hat{\underline{x}} + \hat{\mathbf{B}}\underline{w} \tag{3.40}$$

To obtain this form, except in the velocity and attitude states, the navigation states can be subtracted from the perturbed design states to yield:

$$\delta\dot{\underline{r}}_i^{ned} = \delta\underline{v}_i^{ned} \tag{3.41}$$

$$\delta\dot{\underline{b}}_{a,i} = -\frac{1}{\tau_a}\delta\underline{b}_{a,i}^b + \underline{w}_{a,i} \tag{3.42}$$

$$\delta\dot{\underline{b}}_{g,i} = -\frac{1}{\tau_g}\delta\underline{b}_{g,i}^b + \underline{w}_{g,i} \tag{3.43}$$

The velocity and attitude states require more algebraic manipulation but, in the end, can now be represented as:

$$\delta\dot{\underline{v}}_i^{ned} = -\mathbf{R}_{\hat{b},i}^{ned}(\tilde{\underline{f}}^b - \hat{\underline{b}}_{a,i}) \times \delta\underline{\theta}_{b,i} - \mathbf{R}_{\hat{b},i}^{ned}\delta\underline{b}_{a,i}^b - \mathbf{R}_{\hat{b},i}^{ned}\underline{n}_{f,i} \tag{3.44}$$

$$\delta\dot{\underline{\theta}}_{b,i} = -(\tilde{\underline{\omega}}^b - \hat{\underline{b}}_{g,i}) \times \delta\underline{\theta}_{b,i} - \delta\underline{b}_{g,i}^b + \underline{n}_{\omega,i} \tag{3.45}$$

where the quaternion has been replaced by vector of error angles, $\delta\underline{\theta}_{b,i}$. This detail makes this filter a Multiplicative Extended Kalman Filter (MEKF) and reduces the dimension of the covariance matrix to $15n \times 15n$ instead of $16n \times 16n$. The propagation of the covariance can now be calculated as:

$$\dot{\hat{\mathbf{P}}} = \hat{\mathbf{F}}\hat{\mathbf{P}} + \hat{\mathbf{P}}\hat{\mathbf{F}}^T + \hat{\mathbf{B}}\hat{\mathbf{Q}}\hat{\mathbf{B}}^T \tag{3.46}$$

where

$$\mathbf{F}_i = \begin{bmatrix} 0_{3\times3} & \mathbf{I}_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & -\mathbf{R}_{\hat{b},i}^{ned}(\tilde{\underline{f}}_i^b - \hat{\underline{b}}_{a,i})\times & -\mathbf{R}_{\hat{b},i}^{ned} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & -(\tilde{\underline{\omega}}_i^b - \hat{\underline{b}}_{g,i})\times & 0_{3\times3} & -\mathbf{I}_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -\frac{1}{\tau_a}\mathbf{I}_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -\frac{1}{\tau_g}\mathbf{I}_{3\times3} \end{bmatrix} \tag{3.47}$$

$$\mathbf{B}_i = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \mathbf{R}_{\hat{b},i}^{ned} \\ 0_{3\times3} & \mathbf{I}_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & \mathbf{I}_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix} \tag{3.48}$$

and where, in the centralized filter architecture,

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_i & 0_{15\times15} & \dots \\ 0_{15\times15} & \mathbf{F}_{i+1} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix} \tag{3.49}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_i & 0_{15\times15} & \dots \\ 0_{15\times15} & \mathbf{B}_{i+1} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}. \tag{3.50}$$

In a federated architecture, equation 3.46 would be computed $n$ separate times using $\mathbf{F}_i$ and $\mathbf{B}_i$. This effectively decouples the solutions and does not provide a master covariance matrix. This was done in the author's previous work [34] and shown to produce equivalent results to the centralized filter when the assumption of independent measurements is not violated.

### 3.6 Position Measurements

The position measurement, provided by GPS, is processed at a slower rate (1 Hz) than the IMU measurements which are being treated as continuous but are actually available at 100 Hz. The filter is constructed to handle the case where there is only one GPS receiver and thus the measurement shared among the IMUs. Since the measurement only encompasses position, it can be represented by

$$\underline{\tilde{z}} = \underline{r}^{ned} + \underline{\nu} \tag{3.51}$$

where $\underline{\nu}$ is random Gaussian white noise with standard deviation $\sigma_m$. The measurement is related to the position of each IMU by

$$\underline{r}_i^{ned} = \underline{r}^{ned} + \mathbf{R}_{\hat{b},i}^{ned}\underline{l}_i \tag{3.52}$$

Combining the previous two equations, the measurement for each IMU can be written as

$$\underline{z}_i = \underline{r}_i^{ned} - \mathbf{R}_{\hat{b},i}^{ned}\underline{l}_i + \underline{\nu} = h(\underline{x}_{true}) + \underline{\nu} \tag{3.53}$$

Noteworthy is that the noise of the GPS measurement observed by each IMU filter is the same, implying that the off-diagonal terms of the error covariance matrix will be non-zero. A perturbation approach is taken to obtain the linearized measurement equation, just as with the dynamics equations in the previous section. The desired form of the measurement equation is

$$\delta\underline{z}_i = \mathbf{H}\delta\underline{x}_i + \underline{\nu} \tag{3.54}$$

Proceeding as before:

$$\underline{z}_i = \hat{\underline{z}}_i + \delta\underline{z}_i = \hat{\underline{r}}_i^{ned} + \delta\underline{r}_i^{ned} - [(\mathbf{I} - (\delta\underline{\theta}_{b,i}\times))\mathbf{R}_{ned}^{\hat{b},i}]^T\underline{l}_i + \underline{\nu} \tag{3.55}$$

$$\hat{\underline{z}}_i + \delta\underline{z}_i = \hat{\underline{r}}_i^{ned} + \delta\underline{r}_i^{ned} - \mathbf{R}_{\hat{b},i}^{ned}(\mathbf{I} + (\delta\underline{\theta}_{b,i}\times))\underline{l}_i + \underline{\nu} \tag{3.56}$$

$$\hat{\underline{z}}_i + \delta\underline{z}_i = \hat{\underline{r}}_i^{ned} + \delta\underline{r}_i^{ned} - \mathbf{R}_{\hat{b},i}^{ned}\underline{l}_i - \mathbf{R}_{\hat{b},i}^{ned}(\delta\underline{\theta}_{b,i}\times)\underline{l}_i + \underline{\nu} \tag{3.57}$$

$$\hat{\underline{z}}_i + \delta\underline{z}_i = \hat{\underline{r}}_i^{ned} + \delta\underline{r}_i^{ned} - \mathbf{R}_{\hat{b},i}^{ned}\underline{l}_i + \mathbf{R}_{\hat{b},i}^{ned}(\underline{l}_i\times)\delta\underline{\theta}_{b,i} + \underline{\nu} \tag{3.58}$$

$$\delta\underline{z}_i = \delta\underline{r}_i^{ned} + \mathbf{R}_{\hat{b},i}^{ned}(\underline{l}_i\times)\delta\underline{\theta}_{b,i} + \underline{\nu} \tag{3.59}$$

The linearized measurement equation is therefore

$$\delta\bar{z} = \begin{bmatrix} \delta\underline{r}_1^{ned} + \mathbf{R}_{\hat{b},1}^{ned}(\underline{l}_1\times)\delta\underline{\theta}_{b,1} + \underline{\nu}_1 \\ \delta\underline{r}_2^{ned} + \mathbf{R}_{\hat{b},2}^{ned}(\underline{l}_2\times)\delta\underline{\theta}_{b,2} + \underline{\nu}_1 \\ \vdots \end{bmatrix} \tag{3.60}$$

or

$$\delta\bar{z} = \begin{bmatrix} \hat{\mathbf{H}}_1 & 0_{3\times15} & \cdots \\ 0_{3\times15} & \hat{\mathbf{H}}_2 & \cdots \\ \vdots & \vdots & \hat{\mathbf{H}}_N \end{bmatrix} \delta\underline{x} + \begin{bmatrix} \mathbf{I}_{3\times3} \\ \mathbf{I}_{3\times3} \\ \vdots \end{bmatrix} \underline{\nu} \tag{3.61}$$

where

$$\hat{\mathbf{H}}_i = \begin{bmatrix} I_{3\times3} & 0_{3\times3} & \mathbf{R}_{\hat{b},i}^{ned}(\underline{l}_i\times) & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{3.62}$$

Note that $\underline{\nu}$ in this formulation is mapped into each measurement, which necessarily means that the state estimates and covariance matrix will exhibit correlations after every update step. This accounts for the fact that only one GPS antenna is present on the aircraft, a requirement that can be relaxed simply by rewriting equation 3.61 as

$$\delta\bar{z} = \begin{bmatrix} \hat{\mathbf{H}}_1 & 0_{3\times15} & \cdots \\ 0_{3\times15} & \hat{\mathbf{H}}_2 & \cdots \\ \vdots & \vdots & \hat{\mathbf{H}}_N \end{bmatrix} \delta\underline{x} + \begin{bmatrix} \underline{\nu}_1 \\ \underline{\nu}_2 \\ \vdots \end{bmatrix} \tag{3.63}$$

where each $\underline{\nu}_i$ represents a unique vector of random Gaussian white noise with standard deviation $\sigma_m$. This distinction is important for the sensor fusion derivations in chapter 4.

### 3.7 State and Covariance Update

The *a priori* estimate of the error state vector is assumed to be zero for all Kalman update times. Given the current state estimate:

$$\hat{\underline{\tilde{z}}}_k = \hat{\underline{r}}_k^{ned} \tag{3.64}$$

The error state, or *a posteriori*, estimate can be computed as

$$\delta\underline{x}_k^+ = \hat{\mathbf{K}}_k(\underline{\tilde{z}}_k - \hat{\underline{\tilde{z}}}_k) \tag{3.65}$$

where $\underline{\tilde{z}}_k$ is the measurement at time $t_k$ and the Kalman Gain is

$$\hat{\mathbf{K}}_k = \hat{\mathbf{P}}_k^- \hat{\mathbf{H}}_k^T (\hat{\mathbf{H}}_k \hat{\mathbf{P}}_k^- \hat{\mathbf{H}}_k^T + \hat{\mathbf{R}})^{-1} \tag{3.66}$$

The strength of the measurement noise, $\hat{\mathbf{R}}$ is a constant, diagonal matrix of measurement noise that is either specified by the IMU manufacturer or otherwise known. In the centralized case that the measurement is shared between the individual INS filters,

$$\hat{\mathbf{R}} = \mathbf{G}\mathbf{R}\mathbf{G}^T \tag{3.67}$$

where

$$\mathbf{G} = \begin{bmatrix} \mathbf{I}_{3\times 3} \\ \mathbf{I}_{3\times 3} \\ \vdots \end{bmatrix}_{3n\times 3}. \tag{3.68}$$

Otherwise, $\hat{\mathbf{R}}$ is block diagonal. The covariance estimate is updated using

$$\hat{\mathbf{P}}_k^+ = (\mathbf{I} - \hat{\mathbf{K}}_k \hat{\mathbf{H}}_k)\hat{\mathbf{P}}_k^- (\mathbf{I} - \hat{\mathbf{K}}_k \hat{\mathbf{H}}_k)^T + \hat{\mathbf{K}}_k \hat{\mathbf{R}} \hat{\mathbf{K}}_k^T. \tag{3.69}$$

This is the Jospeph form of the standard covariance update equation. The states can then be updated by adding the error state estimates to the previous state estimates as follows:

$$(\underline{r}_i^{ned})_c = \underline{r}_i^{ned} + (\delta \underline{r}_i^{ned})^+ \tag{3.70}$$

$$(\underline{v}_i^{ned})_c = \underline{v}_i^{ned} + (\delta \underline{v}_i^{ned})^+ \tag{3.71}$$

$$(q_{ned}^b)_c = (\delta q_{\hat{b}}^b)^+ \otimes q_{ned}^{\hat{b}} = \begin{bmatrix} \frac{(\delta \underline{\theta}_b)^+}{2} \\ 1 \end{bmatrix} \otimes q_{ned}^{\hat{b}} \tag{3.72}$$

$$(\underline{b}_{a,i}^{ned})_c = \underline{b}_{a,i}^{ned} + (\delta \underline{b}_{a,i}^{ned})^+ \tag{3.73}$$

$$(\underline{b}_{g,i}^{ned})_c = \underline{b}_{g,i}^{ned} + (\delta \underline{b}_{g,i}^{ned})^+. \tag{3.74}$$

With the measurement update defined, the simulation is run with position updates occurring every one second.

A summary of the numeric values used in the testing of the filter is provided in Table 3.1. The values used for testing the algorithm were the same between IMUs, so only one set of specifications is provided. The noise density and bias offset values for a MEMS IMU were adapted from [35].

### 3.8 EKF Verification

The state and covariance results for position, velocity, attitude, accelerometer biases, and gyroscope biases generated using the EKF described in the previous section for two identical IMUs with uncorrelated measurements are given in Figs. 3.2-3.6. With 200 Monte Carlo realizations, it can be noted that, in all but a few cases, the $3\sigma$ variances align very well with the predicted variances generated via the covariance propagation. The down direction in velocity and accelerometer bias are exceptions, and the reasons for this are unknown to the author. When the GPS updates are shared between IMUs and the covariance becomes correlated, the results still behave well but exhibit anomalous tendencies. These results are shown in Appendix B with comments.

Table 3.1: EKF Parameters used in verification and testing

| Description | Symbol | Value | Units |
|---|---|---|---|
| Accelerometer ECRV time constant | $\tau_a$ | 60 | s |
| Gyroscope ECRV time constant | $\tau_g$ | 60 | s |
| Steady state accelerometer bias $1\sigma$ | $\sigma_{ss,a}$ | 0.019613 | $m/s^2$ |
| Steady state gyroscope bias $1\sigma$ | $\sigma_{ss,g}$ | 0.001212 | rad/s |
| Velocity random walk | vrw | 0.001177 | $m/s^2/\sqrt{Hz}$ |
| Angular random walk | arw | $4.3633 \times 10^{-4}$ | $rad/s/\sqrt{Hz}$ |
| Initial position $1\sigma$, N | $\sigma_{r,N}$ | 1.0 | m |
| Initial position $1\sigma$, E | $\sigma_{r,E}$ | 1.0 | m |
| Initial position $1\sigma$, D | $\sigma_{r,D}$ | 3.0 | m |
| GPS measurement $1\sigma$, N | $\sigma_{m,N}$ | 1.0 | m |
| GPS measurement $1\sigma$, E | $\sigma_{m,E}$ | 1.0 | m |
| GPS measurement $1\sigma$, D | $\sigma_{m,D}$ | 3.0 | m |
| Initial velocity $1\sigma$, (NED) | $\sigma_{v,N}$,$\sigma_{v,E}$,$\sigma_{v,D}$ | 0.1 | m/s |
| Initial angular $1\sigma$ (roll,pitch,yaw) | $\sigma_{\theta,1}$,$\sigma_{\theta,2}\sigma_{\theta,3}$ | 0.1 | rad |
| Initial accelerometer $1\sigma$ X,Y,Z | $\sigma_{a,x}, \sigma_{a,y}, \sigma_{a,z}$ | 0.001 | g |
| Initial gyroscope $1\sigma$ (roll,pitch,yaw) | $\sigma_{g,1}, \sigma_{g,2}, \sigma_{g,3}$ | 1.0 | deg/hr |
| Frequency of IMU measurements | $1/\mathrm{dt}_{IMU}$ | 100 | Hz |
| Frequency of GPS measurements | $1/\mathrm{dt}_{GPS}$ | 1 | Hz |
| Gravity, D | $g$ | 9.8158 | $m/s^2$ |

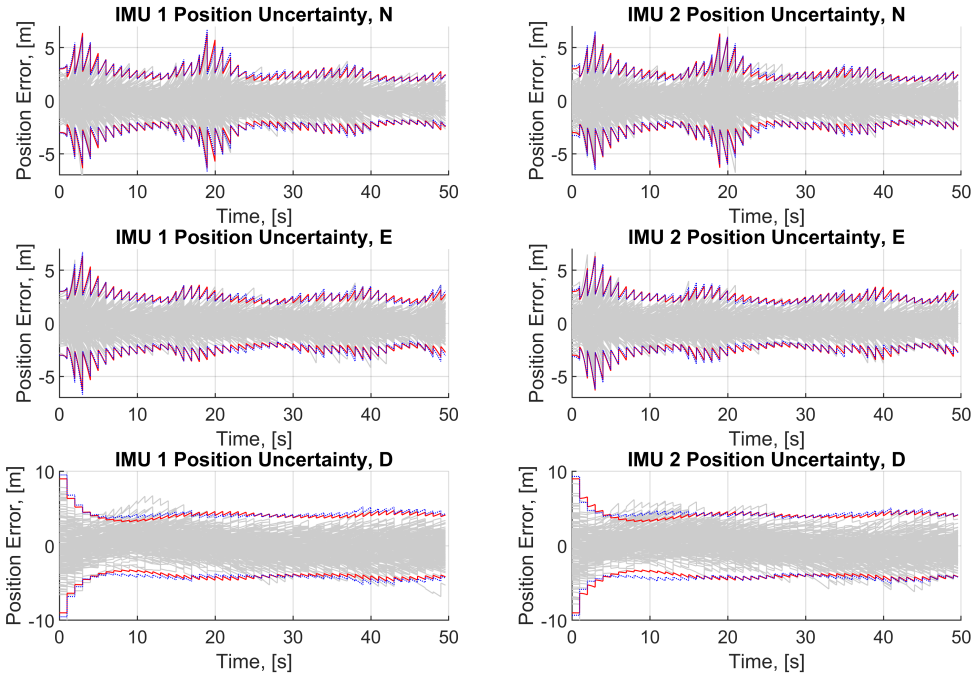


Fig. 3.2: State and covariance estimates for position, 200 Monte Carlo runs.

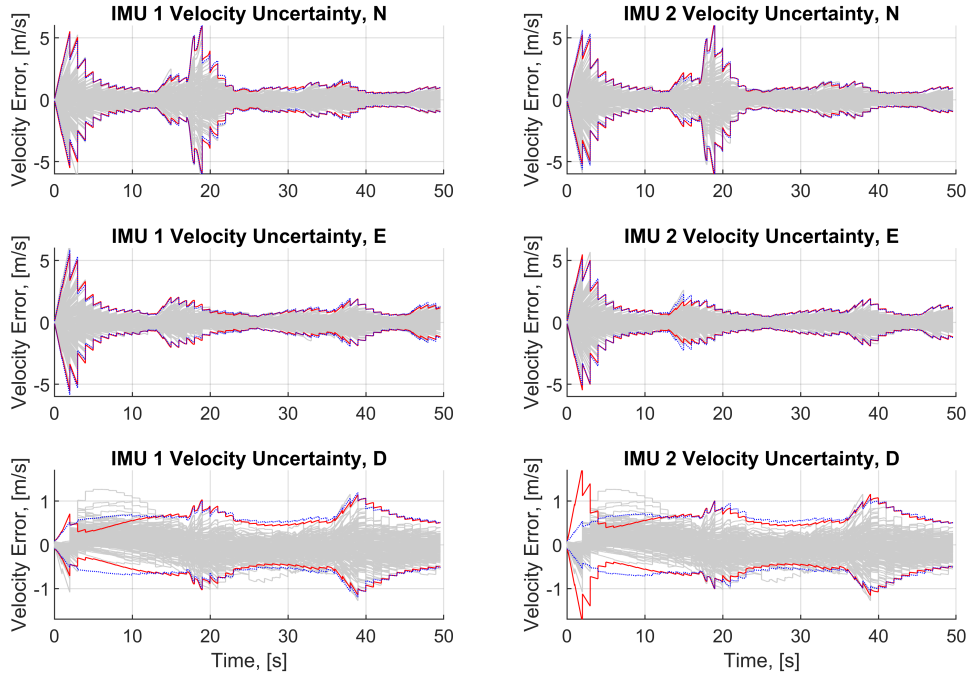


Fig. 3.3: State and covariance estimates for velocity, 200 Monte Carlo runs.

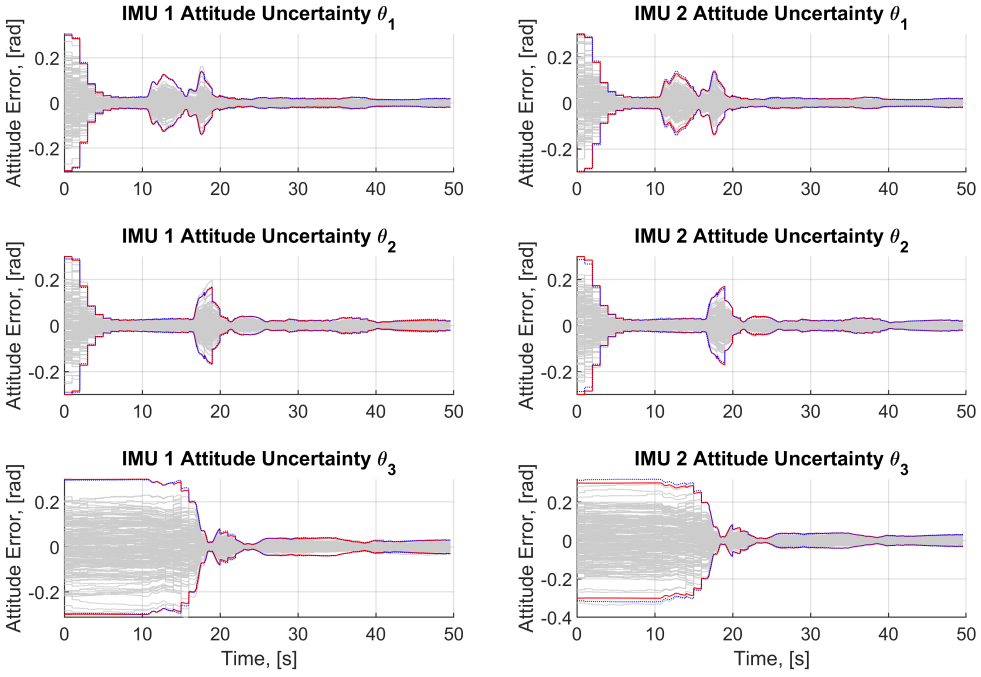


Fig. 3.4: State and covariance estimates for attitude, 200 Monte Carlo runs.

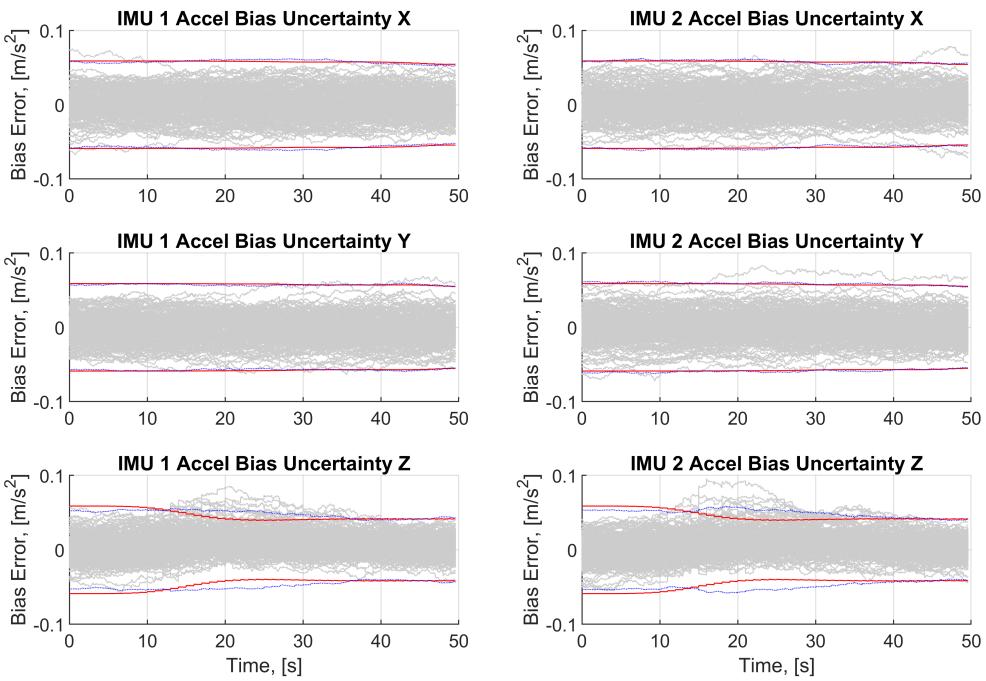


Fig. 3.5: State and covariance estimates for accelerometer biases, 200 Monte Carlo runs.

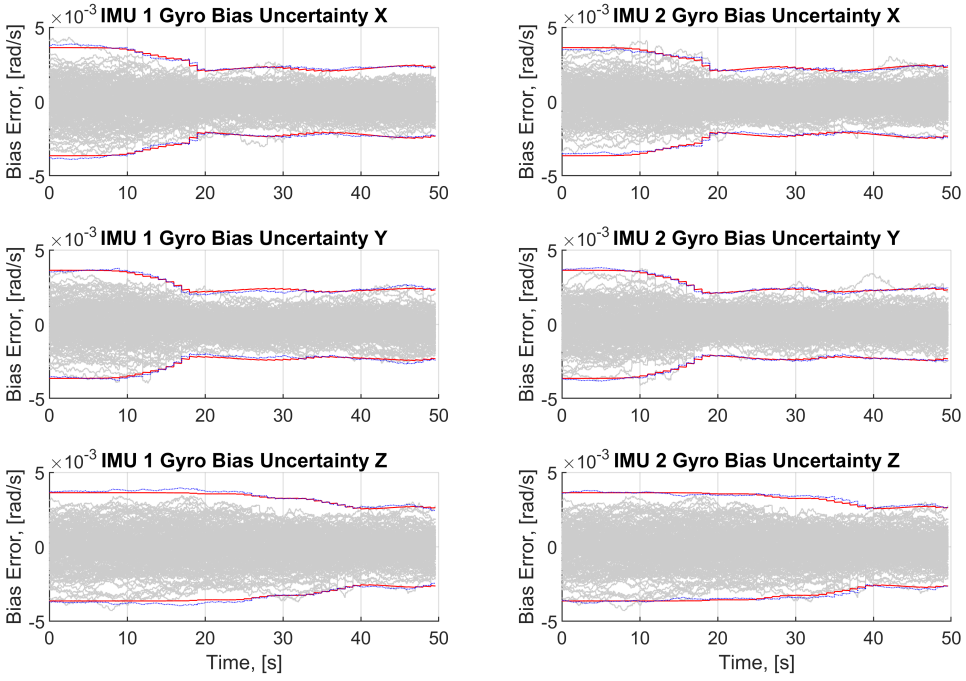


Fig. 3.6: State and covariance estimates for gyroscope biases, 200 Monte Carlo runs.

# CHAPTER 4

# OBSERVATION DOMAIN FUSION

In order to investigate the potential advantages of the distributed geometry of the IMUs, a gyroscope-free, observation domain formulation was implemented in MATLAB. This particular algorithm was succinctly stated in [8], based on work by [36] and others, and is shown again here. Additionally, a maximum likelihood multi-IMU approach was implemented.

## 4.1 Gyro-Free Least Squares Estimator

This gyroscope-free method, which amounts to a least-squares estimator for angular rate $\hat{\underline{\omega}}$ and specific force $\hat{\underline{f}}$, begins by rewriting equation 2.1 as

$$\underline{f}_i - \underline{f} = (\mathbf{\Omega}^2 + \dot{\mathbf{\Omega}})\underline{r}_i \tag{4.1}$$

where $\dot{\mathbf{\Omega}}$ and $\mathbf{\Omega}$ are the skew-symmetric cross product matrices of $\dot{\underline{\omega}}$ and $\underline{\omega}$ respectively. Rearranging the the angular acceleration cross product changes its sign and yields

$$\underline{f}_i - \underline{f} = \mathbf{\Omega}^2\underline{r}_i - \mathbf{\Omega}_{r_i}\dot{\underline{\omega}}. \tag{4.2}$$

The angular acceleration tensor $\mathbf{W}$ can now be defined

$$\mathbf{W}\underline{r}_i = (\mathbf{\Omega}^2 + \dot{\mathbf{\Omega}})\underline{r}_i = \mathbf{\Omega}^2\underline{r}_i - \mathbf{\Omega}_{r_i}\dot{\underline{\omega}}. \tag{4.3}$$

Defining each measurement to be $\underline{y}_i$ with noise $\underline{n}_i$,

$$\underline{y}_i = \underline{f}_i + \underline{n}_i \tag{4.4}$$

the measurement equation can be written for $n$ triads:

$$\mathbf{Y} = \mathbf{XR} + \mathbf{N} \tag{4.5}$$

where the matrices are defined as

$$\mathbf{Y} = \begin{bmatrix} \underline{y}_1 & \dots & \underline{y}_n \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} \underline{f} & \mathbf{W} \end{bmatrix} \tag{4.6}$$

$$\mathbf{R} = \begin{bmatrix} 1 & \dots & 1 \\ \underline{r}_1 & \dots & \underline{r}_n \end{bmatrix} \quad \mathbf{N} = \begin{bmatrix} \underline{n}_1 & \dots & \underline{n}_n \end{bmatrix} \tag{4.7}$$

$\mathbf{W}$ only has six degrees of freedom, meaning that although there are nine parameters in the $3 \times 3$ matrix being estimated, there are only six independent quantities representing $\dot{\underline{\omega}}$ and $\underline{\omega}$. If this fact is ignored [8], the least-square estimate of $\mathbf{X}$ is simply

$$\hat{\mathbf{X}} = \mathbf{YR}^T(\mathbf{RR}^T)^{-1} \tag{4.8}$$

Since $\mathbf{X}$ necessarily has dimensions of $3 \times 4$, four accelerometer triads are needed. For $\mathbf{R}$ to be full rank, the geometry between the sensors must span three dimensional space. Now that there is an estimate of $\hat{\mathbf{W}}$, the angular acceleration and velocity can be calculated as

$$\hat{\dot{\underline{\omega}}} = \begin{bmatrix} \hat{w}_{3,2} - \hat{w}_{2,3} \\ \hat{w}_{1,3} - \hat{w}_{3,1} \\ \hat{w}_{2,1} - \hat{w}_{1,2} \end{bmatrix} \tag{4.9}$$

where $\hat{w}_{ij}$ are elements of $\hat{\mathbf{W}}$. The angular velocity is

$$\hat{\underline{\omega}}\hat{\underline{\omega}}^T = \frac{1}{2}(\hat{\mathbf{W}} + \hat{\mathbf{W}}^T) - \frac{1}{4}\mathrm{tr}(\hat{\mathbf{W}} + \hat{\mathbf{W}}^T)\mathbf{I}_{3\times 3}. \tag{4.10}$$

Because this equation is quadratic, sign of the angular velocity is ambiguous. Given that the multi-IMU system being considered in this research includes gyroscopes, the sign of

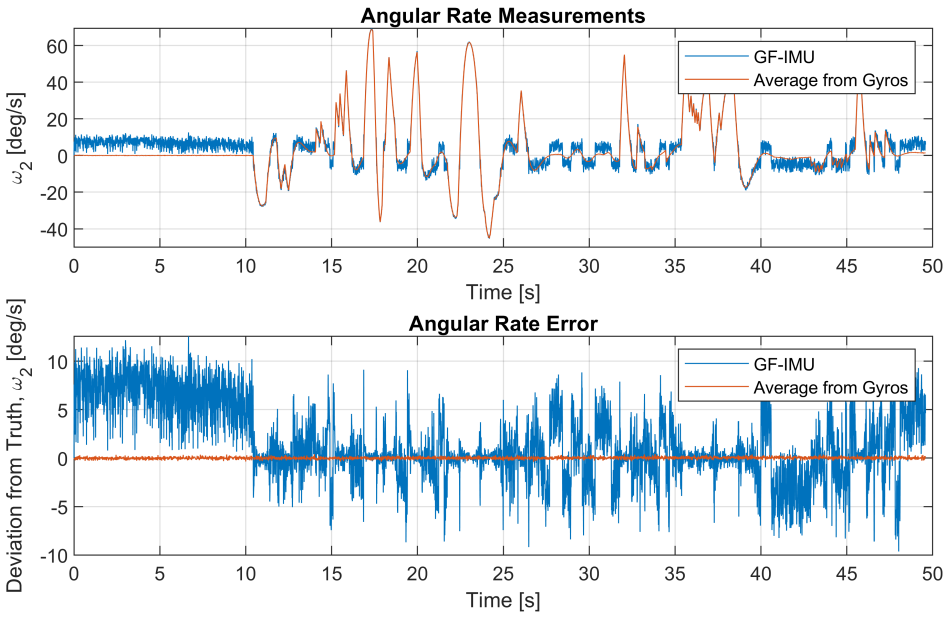


Fig. 4.1: Angular rate ($\hat{\underline{\omega}}_2$) estimation using the GF algorithm and the true angular rate to resolve the sign ambiguity.

the angular rates can be determined. Also noteworthy is that there must be at least four gyroscope triads because $\mathbf{X}$ is necessarily $3 \times 4$. Fig. 4.1 shows the typical results of the gyro-free least-squares algorithm using five MEMS accelerometer triads and where the true angular rate vector is used to resolve the sign ambiguity. This shows that, using MEMS accelerometers and gyros as specified in Table 3.1, the algorithm yields poor performance even with the sign ambiguity known perfectly. An initial, noisy bias is also present before the motion begins. The source of this is currently unknown, but it seems to contain all of the uncertainty during that period of time. The other components of angular acceleration are identically zero until motion begins at approximately 10 seconds.

Clearly, this gyroscope-free algorithm, described as state-of-the-art in [8], is not suited for MEMS IMUs in the context of sUAS navigation. In practice, a real navigation system would not have access to the true angular rate, so the resolution of the sign ambiguity becomes more noisy when an average angular rate, based on measurements from all of the IMUs' gyroscopes, is used instead. This is shown in Fig. 4.2. Interestingly, the increased sign uncertainty actually removes the aforementioned bias.

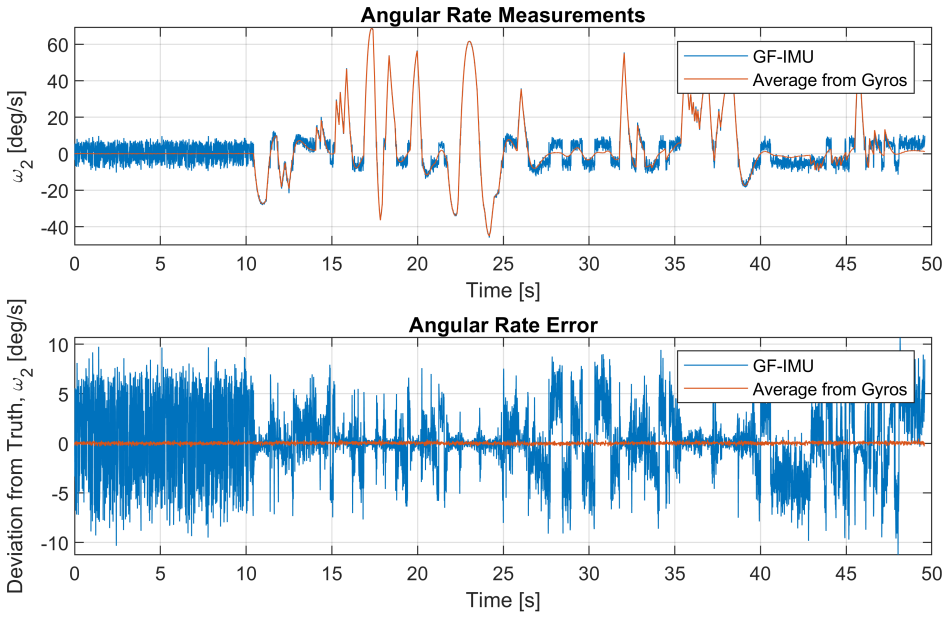


Fig. 4.2: Angular rate ($\hat{\underline{\omega}}_2$) estimation using the GF algorithm and an average angular rate to resolve the sign ambiguity.

## 4.2 Maximum Likelihood Estimator

As an alternative to the least-squares estimator, [8] also proposed and analyzed a maximum likelihood estimator that uses all available gyroscope and accelerometer measurements. This estimator produces angular velocity measurements that are numerically equivalent to the average of the individual gyroscope triad angular velocities at lower angular rates. It is important to note:

1. The average is weighted by process noise specifications given in a predetermined covariance matrix, $\mathbf{Q}_w$. This is defined as in Chapter 3 but without the bias terms.

2. The accelerometers are incorporated into the estimate at high angular rates, significantly increasing the measurement range of the sensor package to well beyond gyroscope saturation limits.

3. A direct estimate of the angular acceleration is produced, which could be useful in some applications.

The maximum likelihood estimator can be broken into two main steps: the estimation of $\hat{\underline{\omega}}$ and the estimation of $\hat{\underline{\phi}}$ where

$$\hat{\underline{\phi}} = \begin{bmatrix} \hat{\underline{\omega}} \\ \hat{\underline{f}} \end{bmatrix} \tag{4.11}$$

An iterative (Gauss-Newton) method is given for solving for $\hat{\underline{\omega}}$ at time $t_k$,

$$\hat{\underline{\omega}}_{k+1} = \hat{\underline{\omega}}_k + (\mathbf{J}_h^T \mathbf{P} \mathbf{J}_h) \mathbf{J}_h^T \mathbf{P} (\underline{y} - \underline{h}(\hat{\underline{\omega}}_k)), \tag{4.12}$$

where $\hat{\underline{\omega}}_0$ is given by a weighted least-squares estimate of the angular velocity read by the gyroscopes,

$$\hat{\underline{\omega}}_0 = ((\mathbf{1}_n^T \otimes \mathbf{I}_{3\times 3}) \mathbf{Q}_\omega^{-1} (\mathbf{1}_n \otimes \mathbf{I}_{3\times 3}))^{-1} (\mathbf{1}_n \otimes \mathbf{I}_{3\times 3}) \mathbf{Q}_\omega^{-1} \underline{y}_\omega \tag{4.13}$$

$\mathbf{J}_h$ is the Jacobian of the nonlinear part of the inertial sensor array model, $\underline{h}(\underline{\omega})$,

$$\mathbf{J}_h = \begin{bmatrix} (\mathbf{\Omega}_{\omega_k}^T \mathbf{\Omega}_{r_1} + \mathbf{\Omega}_{\omega_k \times r_1})^T & (\mathbf{\Omega}_{\omega_k}^T \mathbf{\Omega}_{r_1} + \mathbf{\Omega}_{\omega_k \times r_1})^T & \cdots & \mathbf{1}_n \otimes \mathbf{I}_{3\times 3} \end{bmatrix}^T. \tag{4.14}$$

$$\underline{h}(\hat{\underline{\omega}}) = \begin{bmatrix} \mathbf{\Omega}_{\hat{\omega}_k}^2 \underline{l}_1 \\ \vdots \\ \mathbf{1}_n \otimes \hat{\underline{\omega}}_k \end{bmatrix}, \tag{4.15}$$

and $\mathbf{Q}_\omega$ is the partition of $\mathbf{Q}_w$ that contains the gyroscope measurement variances. $\mathbf{P}$ is given by

$$\mathbf{P} = \mathbf{Q}_w^{-1} - \mathbf{Q}_w^{-1} \mathbf{H} (\mathbf{H}^T \mathbf{Q}_w^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{Q}_w^{-1} \tag{4.16}$$

where $\mathbf{H}$ is related to the angular acceleration geometry

$$\mathbf{H} = \begin{bmatrix} \mathbf{G} & \mathbf{1}_n \otimes \mathbf{I}_{3\times 3} \end{bmatrix} \tag{4.17}$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{\Omega}_{r_1} \\ \vdots \\ \mathbf{\Omega}_{r_n} \end{bmatrix} \tag{4.18}$$

and the accelerometer and gyroscope measurements are passed in via

$$\underline{y} = \begin{bmatrix} \underline{f}_1 \\ \underline{f}_2 \\ \vdots \\ \underline{f}_n \\ \underline{\omega}_1 \\ \underline{\omega}_2 \\ \vdots \\ \underline{\omega}_n \end{bmatrix}. \tag{4.19}$$

Note that the $\otimes$ symbol denotes the Kronecker product. Once $\hat{\underline{\omega}}$ is calculated using equation 4.12, $\hat{\underline{\phi}}$ is found using

$$\hat{\underline{\phi}} = (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})\mathbf{H}^T\mathbf{Q}_w^{-1}(\underline{y} - \underline{h}(\hat{\underline{\omega}})) \tag{4.20}$$

While this algorithm has its aforementioned advantages, at the angular rates a sUAS will experience, it results in angular velocity information that can be readily obtained using a simple, less-computationally intensive average of gyroscope measurements. Another potential disadvantage in this application is shown in Fig. 4.3, which shows an estimate of specific force over time at the origin of the body frame of the aircraft. Clearly, this result is not zero-mean and, when used as an input to the navigation EKF as described in Chapter 3, it produces drifts in the position and velocity solutions. This tendency persists when more IMUs are added to the simulation. One possible solution to this would be to place one IMU at the origin of the body frame and use its specific force measurements exclusively, but this would partially defeat the purpose of having distributed accelerometers in the first place.

### 4.3 Improving Maximum Likelihood Angular Velocity Estimates

One quantity that is generated in the previous algorithm is an estimate of the angular acceleration. It is conceivable that this quantity could be integrated in order to generate a

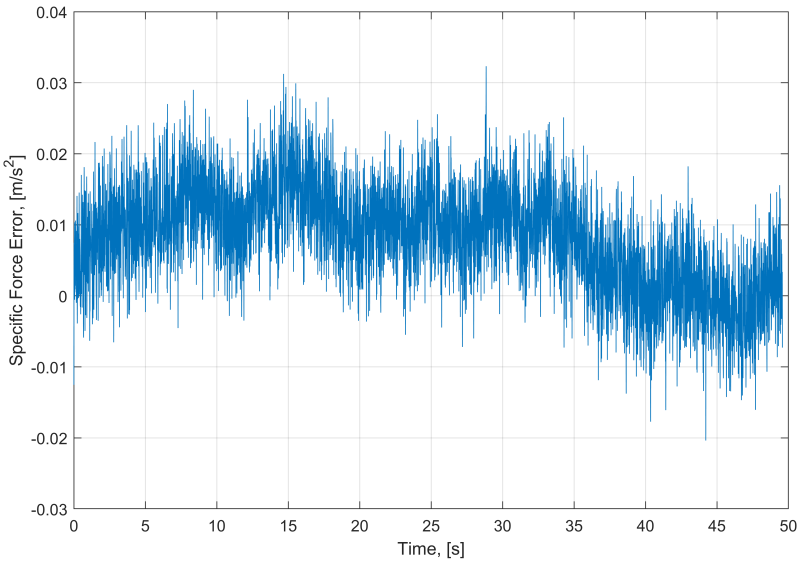


Fig. 4.3: Specific force estimate produced by maximum likelihood estimator using 5 identical IMUs at 1 meter lever arms.

second estimate of angular velocity, one that would likely increase in reliability when the rotational dynamics of the aircraft are significant. In that spirit, a complementary filter idea was explored wherein the angular velocity estimate of the gyroscope average could be combined with the angular velocity estimate of the integrated angular acceleration. The basic time domain form of the complementary filter can be written in this context as

$$\underline{\hat{\omega}}_f = \alpha \underline{\hat{\omega}}_{int} + (1-\alpha)\underline{\hat{\omega}} \tag{4.21}$$

where $\alpha$ is a weighting coefficient between 0 and 1, $\underline{\hat{\omega}}_f$ is the filtered value of the angular velocity, and $\underline{\hat{\omega}}_{int}$ is the angular velocity generated from trapezoidal integration of the angular acceleration

$$\underline{\hat{\omega}}_{int,k} = \frac{1}{2}(\underline{\hat{\dot{\omega}}}_{k-1} + \underline{\hat{\dot{\omega}}}_k)dt + \underline{\hat{\omega}}_{f,k-1}. \tag{4.22}$$

In order to encompass the stipulation that the integrated solution is more accurate during periods of higher angular motion, $\alpha$ could be defined in a number of ways. One simple way

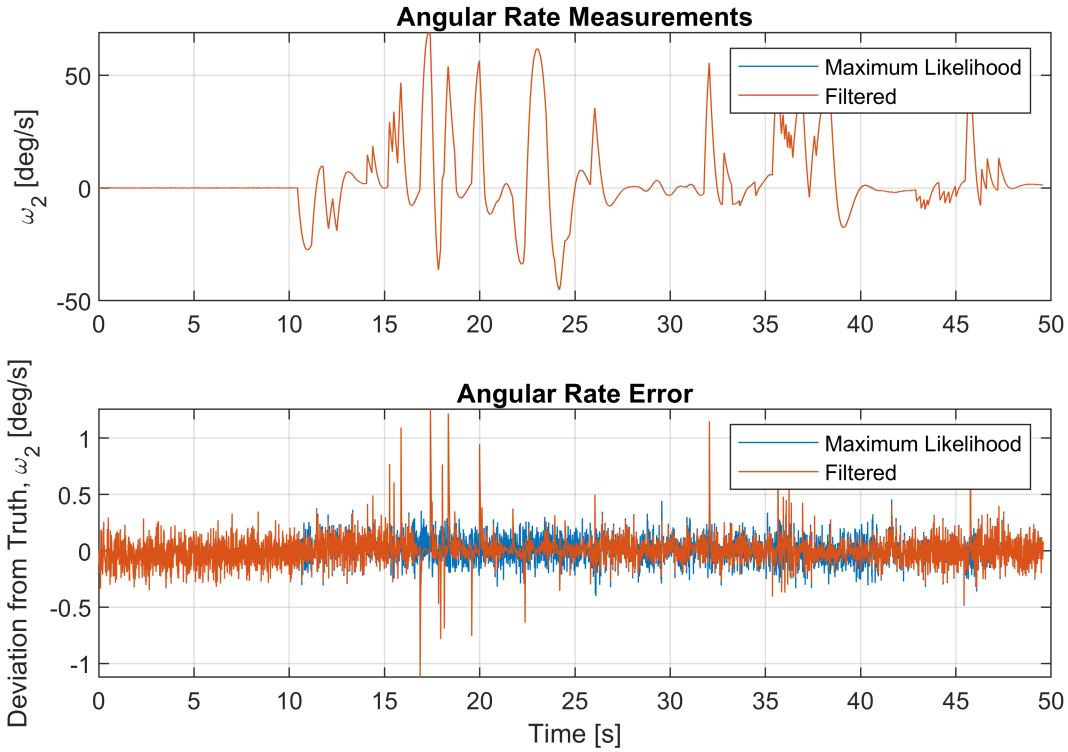


Fig. 4.4: Maximum Likelihood estimate of angular velocity plotted with an estimate of angular velocity derived using the described complementary filter.

would be to relate it to the norm of the angular acceleration the aircraft is experiencing:

$$\alpha = \frac{||\hat{\underline{\omega}}||}{c} \tag{4.23}$$

where $c$ is some constant with the same units as $\hat{\underline{\omega}}$. Using $c = 1.4$, the result of the complementary filter is shown in Fig. 4.4.

The results show that there may indeed be an advantage to incorporating a filtering strategy like this in this context. Continuation of this line of thought is left to future work.

# CHAPTER 5
# ESTIMATION DOMAIN FUSION

## 5.1 Information Conservation Approach

As mentioned in the literature review in Chapter 2, the approach to fusing state estimates calculated by individual INS units into a single (hopefully improved) one is usually done using an information conservation approach. The equations for applying this idea are summarized again here:

$$\mathbf{P}_s^{-1} = \sum_{i=1}^{n} \mathbf{P}_i^{-1} \tag{5.1}$$

$$\hat{\underline{x}}_s = \mathbf{P}_s^{-1} \sum_{i=1}^{n} \mathbf{P}_i^{-1} \hat{\underline{x}}_i. \tag{5.2}$$

where $\mathbf{P}_i$ are the individual state covariance matrices, $\underline{x}_i$ are the individual state estimates, $\mathbf{P}_s$ is the covariance of the combined estimate, and $\underline{x}_s$ is the combined state estimate. This approach works well when used in a federated architecture or in a centralized architecture when the measurement updates are uncorrelated. However, when a GPS observation is used more than once and induces correlations into the larger covariance matrix, this formulation cannot be used.

## 5.2 Weighted Averaging

The deficiency of the information conservation formulation in the centralized filter motivated the derivation of an alternate approach, one that was designed to leverage the correlations present in the covariance matrix. Starting from the idea that the problem can be written as a sum of weighted estimates,

$$\hat{\underline{x}}_s = \mathbf{\Theta}_1 \mathbf{m}_1 \hat{\underline{x}}_1 + \mathbf{\Theta}_2 \mathbf{m}_2 \hat{\underline{x}}_2 + \ldots + \mathbf{\Theta}_i \mathbf{m}_i \hat{\underline{x}}_i = \sum_{i=1}^{n} \mathbf{\Theta}_i \mathbf{m}_i \hat{\underline{x}}_i, \tag{5.3}$$

where $\underline{x}_s$ is "combined" state estimate that is a weighted average of the individual state estimates,

$$\underline{\hat{x}}_s = \begin{bmatrix} \underline{\hat{r}}_s^{ned} \\ \underline{\hat{v}}_s^{ned} \\ \hat{q}_{s,ned}^b \end{bmatrix}. \tag{5.4}$$

An optimization strategy was sought for the optimal values of the weighting matrices, $\mathbf{\Theta}_i$. Since the available covariance is produced for the error states, each weighting matrix will be $9 \times 9$ instead of $10 \times 10$ as would be expected from the dimension of the state vector. This discrepancy can be handled by converting the quaternion to Euler angles before it is weighted and combined. The estimate can then be converted back to a quaternion.

The relative geometry matrix, $\mathbf{m}_i$ accounts for the relative geometry of the position, velocity, and attitude solutions. Since the individual estimates are computed at lever arms, these estimates must be rotated and moved to the orientation and location of a defined point for combination. The true relative position, velocity, and attitude can be represented as

$$\underline{r}^{ned} = \underline{r}_i^{ned} + \mathbf{R}_b^{ned} \underline{l}_i^b \tag{5.5}$$

$$\underline{v}^{ned} = \underline{v}_i^{ned} + \mathbf{R}_b^{ned} (\underline{\omega}_i^b \times \underline{l}_i^b) \tag{5.6}$$

$$\underline{\theta}_b^{ned} = \underline{\theta}_{b_i}^{ned} \tag{5.7}$$

Equations 5.5 and 5.6 are nonlinear and cannot be represented in matrix form. For the purposes of this derivation, it will be assumed that, for the total states, that $\mathbf{m}_i$ exists. For the error states, which will be used later in the derivation,

$$\mathbf{m}_i = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{R}_{\hat{b}}^{ned}(\underline{l}_i^b\times) & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & -\mathbf{R}_{\hat{b}}^{ned}((\underline{\omega}_i^b \times \underline{l}_i^b)\times) & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}. \tag{5.8}$$

A derivation of these terms is provided in Appendix C.

In order to formulate the problem in an optimization context, a constraint was required. Applying the expectation operator to equation 5.3 and simplifying yields

$$\mathrm{E}[\underline{\hat{x}}_s] = \mathrm{E}[\mathbf{\Theta}_1\mathbf{m}_1\underline{\hat{x}}_1 + \mathbf{\Theta}_2\mathbf{m}_2\underline{\hat{x}}_2 + ... + \mathbf{\Theta}_n\mathbf{m}_n\underline{\hat{x}}_n] \tag{5.9}$$

$$\mathrm{E}[\underline{\hat{x}}_s] = \mathbf{\Theta}_1\mathrm{E}[\mathbf{m}_1\underline{\hat{x}}_1] + \mathbf{\Theta}_2\mathrm{E}[\mathbf{m}_2\underline{\hat{x}}_2] + ... + \mathbf{\Theta}_n\mathrm{E}[\mathbf{m}_n\underline{\hat{x}}_n] \tag{5.10}$$

$$\underline{\hat{x}}_s = \mathbf{\Theta}_1\underline{\hat{x}}_s + \mathbf{\Theta}_2\underline{\hat{x}}_s + ... + \mathbf{\Theta}_n\underline{\hat{x}}_s. \tag{5.11}$$

When this equation is factored, an important constraint is obtained:

$$\underline{\hat{x}}_s = (\mathbf{\Theta}_1 + \mathbf{\Theta}_2 + ... + \mathbf{\Theta}_n)\underline{\hat{x}}_s = \left(\sum_{i=1}^{n}\mathbf{\Theta}_i\right)\underline{\hat{x}}_s \tag{5.12}$$

$$\boxed{\sum_{i=1}^{n}\mathbf{\Theta}_i = \mathbf{I}} \tag{5.13}$$

This means that the sum of the weighting matrices is an identity matrix. This is analogous to the scalar case wherein the sum of all weighting factors in an average must equal one.

Since the covariance matrix available from the EKF is valid for the error states and not necessarily the full states, this constraint must be verified for the error states. The approach outlined above does not hold for the error states because

$$\mathrm{E}[\delta\underline{\hat{x}}_s] = \mathrm{E}[\mathbf{m}_i\delta\underline{\hat{x}}_i] = \mathbf{0}_{3\times1} \tag{5.14}$$

resulting in zero divided by zero in the equation analogous to 5.12. This motivates an extension to the previous derivation. A generalized truth state can be defined as

$$\underline{x}_s = \underline{\hat{x}}_s + \delta\underline{x}_s \tag{5.15}$$

and

$$\underline{x}_i = \underline{\hat{x}}_i + \delta\underline{x}_i. \tag{5.16}$$

Substituting these relationships into equation 5.3:

$$(\underline{x}_s - \delta\underline{x}_s) = \sum_{i=1}^{n} \mathbf{\Theta}_i(\underline{x}_i - \delta\underline{x}_i). \tag{5.17}$$

The sum can be split into two sums because $\mathbf{\Theta}_i$ and $\mathbf{m}_i$ are considered static.

$$\underline{x}_s - \delta\underline{x}_s = \sum_{i=1}^{n} \mathbf{\Theta}_i\mathbf{m}_i\underline{x}_i - \sum_{i=1}^{n} \mathbf{\Theta}_i\mathbf{m}_i\delta\underline{x}_i. \tag{5.18}$$

The first summation term is known to be equal to the true state, so it cancels out of the equation on both sides leaving

$$\delta\underline{x}_s = \sum_{i=1}^{n} \mathbf{\Theta}_i\mathbf{m}_i\delta\underline{x}_i. \tag{5.19}$$

which means that the same weighting matrices that are valid for the truth state are valid for the error states and the constraint holds.

### 5.3 Optimal Fusion Method

The objective of the optimal fusion method is to find the optimal values of the individual $\mathbf{\Theta}_i$ matrices that minimize the variances. Mathematically,

$$\mathbf{P}_s = \mathrm{E}[\delta\underline{\hat{x}}\delta\underline{\hat{x}}^T] = \mathbf{\Theta}^T\mathbf{M}\mathbf{P}_{15n\times 15n}\mathbf{M}^T\mathbf{\Theta} \tag{5.20}$$

where

$$\mathbf{\Theta}^T = \begin{bmatrix}\mathbf{\Theta}_1 & \mathbf{\Theta}_2 & \dots & \mathbf{\Theta}_n\end{bmatrix}, \quad \mathbf{\Theta} = \begin{bmatrix}\mathbf{\Theta}_1^T \\ \mathbf{\Theta}_2^T \\ \vdots \\ \mathbf{\Theta}_n^T\end{bmatrix}, \quad \mathbf{M}_{9n\times 15n} = \begin{bmatrix}\mathbf{m}_1 & 0_{9\times 15} & \dots & 0_{9\times 15} \\ 0_{9\times 15} & \mathbf{m}_2 & \dots & 0_{9\times 15} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{9\times 15} & 0_{9\times 15} & \dots & \mathbf{m}_n\end{bmatrix} \tag{5.21}$$

and $\mathbf{P}_{15n\times 15n}$ is known from the Kalman filter. Defining

$$\mathbf{Q} = \mathbf{M}\mathbf{P}_{15n\times 15n}\mathbf{M}^T \tag{5.22}$$

and

$$\mathcal{I}\boldsymbol{\Theta} = \begin{bmatrix} \mathbf{I}_{9\times9} & \mathbf{I}_{9\times9} & \dots & \mathbf{I}_{9\times9} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Theta}_1^T \\ \boldsymbol{\Theta}_2^T \\ \vdots \\ \boldsymbol{\Theta}_n^T \end{bmatrix} = \mathbf{I}_{9\times9} \tag{5.23}$$

and

$$\underline{e}_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \underline{e}_2 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \quad \text{etc.} \tag{5.24}$$

the problem can be stated as minimizing

$$\text{tr}(\mathbf{P}_{9\times9}) = \text{tr}(\boldsymbol{\Theta}^T\mathbf{Q}\boldsymbol{\Theta}) \tag{5.25}$$

such that

$$\mathcal{I}\boldsymbol{\Theta} = \mathbf{I} \tag{5.26}$$

It should be noted that equation 5.26 is an equivalent restatement of equation 5.13. The Lagrangian of the system can be defined as

$$\mathcal{L} = \text{tr}(\boldsymbol{\Theta}^T\mathbf{Q}\boldsymbol{\Theta}) + \sum_{i=1}^{9}\sum_{j=1}^{9} \underline{e}_i^T(\mathcal{I}\boldsymbol{\Theta})\underline{e}_j\lambda_{ij} \tag{5.27}$$

The gradients of this equation must be found. Using a relationship (9) found on page 907 of [37],

$$\frac{\partial}{\partial\boldsymbol{\Theta}}\text{tr}(\boldsymbol{\Theta}^T\mathbf{Q}\boldsymbol{\Theta}) = \mathbf{Q}\boldsymbol{\Theta} + \mathbf{Q}^T\boldsymbol{\Theta} \tag{5.28}$$

The second term in the Lagrangian is a scalar, so

$$\frac{\partial}{\partial\boldsymbol{\Theta}}\underline{e}_i^T\mathcal{I}\boldsymbol{\Theta}\underline{e}_j = \frac{\partial}{\partial\boldsymbol{\Theta}}\text{tr}(\underline{e}_i^T\mathcal{I}\boldsymbol{\Theta}\underline{e}_j) \tag{5.29}$$

The trace has the property that it is equivalent to its cyclic permutations:

$$\frac{\partial}{\partial \mathbf{\Theta}} \text{tr}(\underline{e}_i^T \mathcal{I} \mathbf{\Theta} \underline{e}_j) = \frac{\partial}{\partial \mathbf{\Theta}} \text{tr}(\underline{e}_j \underline{e}_i^T \mathcal{I} \mathbf{\Theta}) \tag{5.30}$$

Another relationship (8) from the same textbook yields

$$\frac{\partial}{\partial \mathbf{\Theta}} \text{tr}(\underline{e}_j \underline{e}_i^T \mathcal{I} \mathbf{\Theta}) = \mathcal{I}^T \underline{e}_i \underline{e}_j^T \tag{5.31}$$

Taken together, the derivative of the Lagrangian can be set equal to zero:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{\Theta}} = (\mathbf{Q} + \mathbf{Q}^T)\mathbf{\Theta} + \sum_{i=1}^{9} \sum_{j=1}^{9} \lambda_{ij} \mathcal{I}^T \underline{e}_i \underline{e}_j^T = 0 \tag{5.32}$$

Solving for $\mathbf{\Theta}$,

$$\mathbf{\Theta} = -(\mathbf{Q} + \mathbf{Q}^T)^{-1} \sum_{i=1}^{9} \sum_{j=1}^{9} \lambda_{ij} \mathcal{I}^T \underline{e}_i \underline{e}_j^T \tag{5.33}$$

The matrix $(\mathbf{Q} + \mathbf{Q}^T)^{-1}$ can be moved inside the sum:

$$\mathbf{\Theta} = -\sum_{i=1}^{9} \sum_{j=1}^{9} \lambda_{ij} (\mathbf{Q} + \mathbf{Q}^T)^{-1} \mathcal{I}^T \underline{e}_i \underline{e}_j^T \tag{5.34}$$

Multiplying both sides by $\mathcal{I}$, the equation reduces to

$$\mathcal{I}\mathbf{\Theta} = -\sum_{i=1}^{9} \sum_{j=1}^{9} \lambda_{ij} \mathcal{I} (\mathbf{Q} + \mathbf{Q}^T)^{-1} \mathcal{I}^T \underline{e}_i \underline{e}_j^T \tag{5.35}$$

Defining

$$\mathbf{R} = \mathcal{I}(\mathbf{Q} + \mathbf{Q}^T)^{-1} \mathcal{I}^T \tag{5.36}$$

and using the relationship previously derived (5.13), yields

$$\mathbf{I} = -\sum_{i=1}^{9} \sum_{j=1}^{9} \lambda_{ij} \mathbf{R} \underline{e}_i \underline{e}_j^T \tag{5.37}$$

which is nine equations with nine unknown $\lambda_{ij}$ terms. Once this system is solved, the Lagrange multipliers can be used to solve for the full matrix $\mathbf{\Theta}$ in equation 5.34.

This derivation is in line with a common approach to solving a correlated scalar random variable weighted average problem, as can be found in [38] (for instance). The application of this idea in the context of INS sensor fusion is believed to be completely novel.

## 5.4 Solving the Optimal Fusion Algorithm

After some manipulation, it was determined that equation 5.37 can be solved as

$$\mathbf{\Lambda} = \mathbf{R}^{-1}\mathcal{I} \tag{5.38}$$

where

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_{11} & \lambda_{12} & \dots \\ \lambda_{21} & \ddots & \vdots \\ \vdots & \dots & \lambda_{99} \end{bmatrix} \tag{5.39}$$

With these values in hand, it is simple to synthesize $\mathbf{\Theta}$ using equation 5.29.

## 5.5 Anomalous Results of Optimal Fusion Method

The previously derived optimal weighted average solution was applied as a static or "snapshot" filtering strategy to the results of the centralized EKF outlined in chapter 3 using two IMUs of identical specifications. For comparison, because the two IMUs were identical, the computation was also done assuming equal weights on each solution. In that case, the $\mathbf{\Theta}$ matrix can be assumed to be

$$\mathbf{\Theta} = 0.5 \begin{bmatrix} \mathbf{I}_{9\times 9} \\ \mathbf{I}_{9\times 9} \end{bmatrix} \tag{5.40}$$

which results in equal, static weights for the position, velocity, and attitude for each IMU. The position results of the naive weighted average approach were shown in [34] and are shown again, along with the results for velocity and attitude, in Figs. 5.1-5.3.

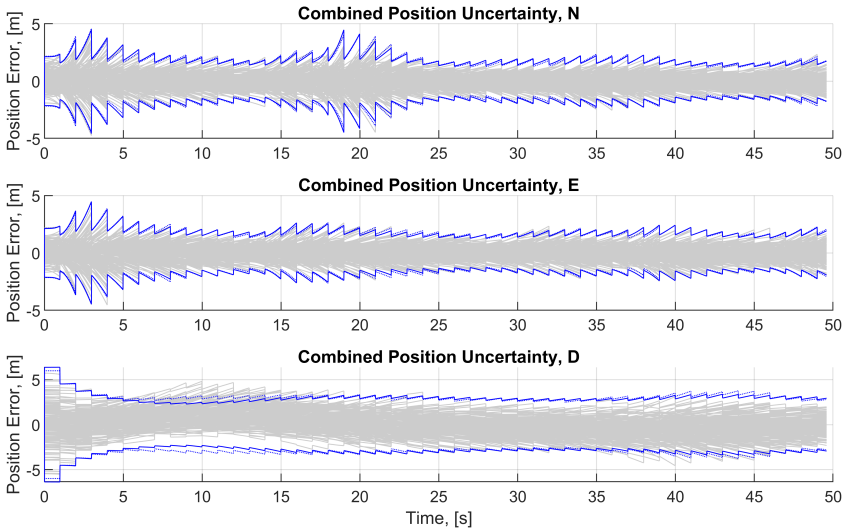


Fig. 5.1: Naive equal fusion method results for position. Grey lines are 200 Monte Carlo realizations, blue dotted lines are ensemble $3\sigma$, blue solid lines are the predicted $3\sigma$.

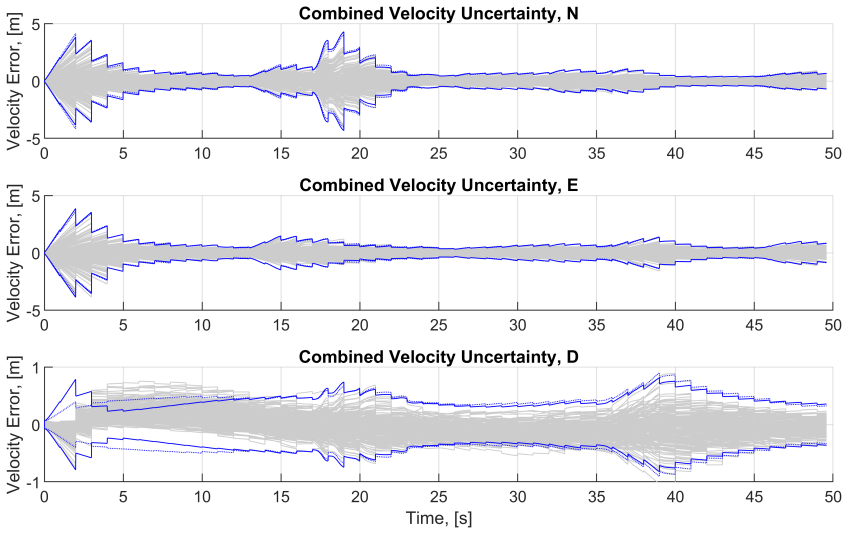


Fig. 5.2: Naive equal fusion method results for velocity. Grey lines are 200 Monte Carlo realizations, blue dotted lines are ensemble $3\sigma$, blue solid lines are the predicted $3\sigma$.

Using the optimal weighted averaging method proposed in section 5.4, the same plots can be generated. Unlike the naive approach, the optimal strategy recalculates the weighting matrix at every iteration and generates unique weights for all components of position, velocity, and attitude for all sensors. Fig. 5.4 shows the position error using the optimal

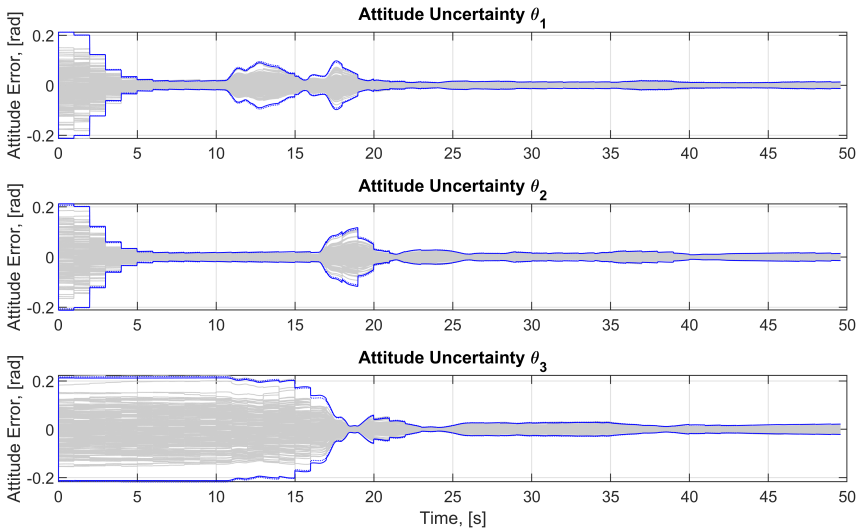


Fig. 5.3: Naive equal fusion method results for attitude. Grey lines are 200 Monte Carlo realizations, blue dotted lines are ensemble $3\sigma$, blue solid lines are the predicted $3\sigma$.

approach.

While the position solution behaves similarly to the naive position solution in Fig. 5.1, the velocity and attitude solutions do not. Figs. 5.5 and 5.6 show these results. There are obvious anomalous tendencies in the velocity and attitude.

## 5.6 Constrained Optimal Fusion Method

Because of the unexplained results and poor performance of the previous optimal fusion algorithm, a modified version is proposed in this section. Instead of allowing the optimization to select separate, individual weighting matrices for each INS's position, velocity, and attitude, which potentially introduced too many degrees of freedom into the system, a constraint was sought such that a unique weighting matrix would be selected for each INS. The position solution, velocity solution, and attitude solution of each INS would be weighted by the same amount. While this sacrifices generality and assumes that the validity of the position, velocity, and attitude solutions from a respective IMU are equal, in exchange it does decrease the computational load. The sum of the weighting matrices $\mathbf{\Theta}_i$ still obeys

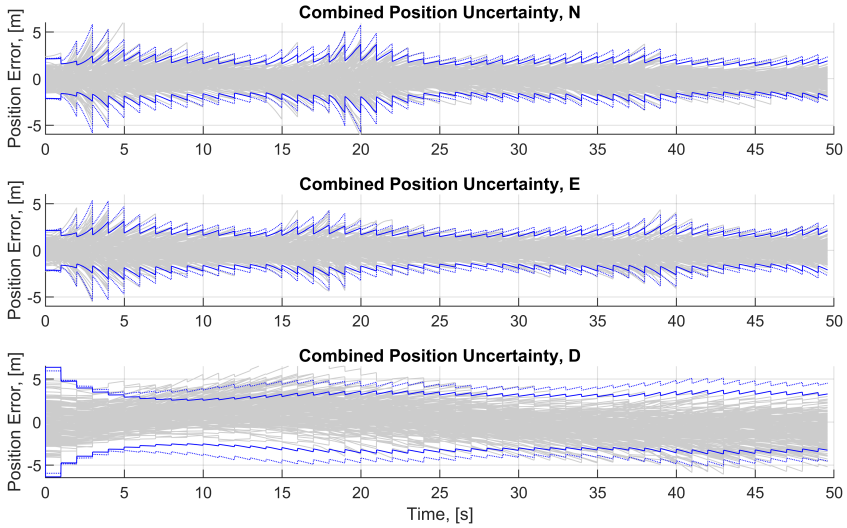


Fig. 5.4: Optimal fusion method results for position. Grey lines are 200 Monte Carlo realizations, blue dotted lines are ensemble $3\sigma$, blue solid lines are the predicted $3\sigma$.

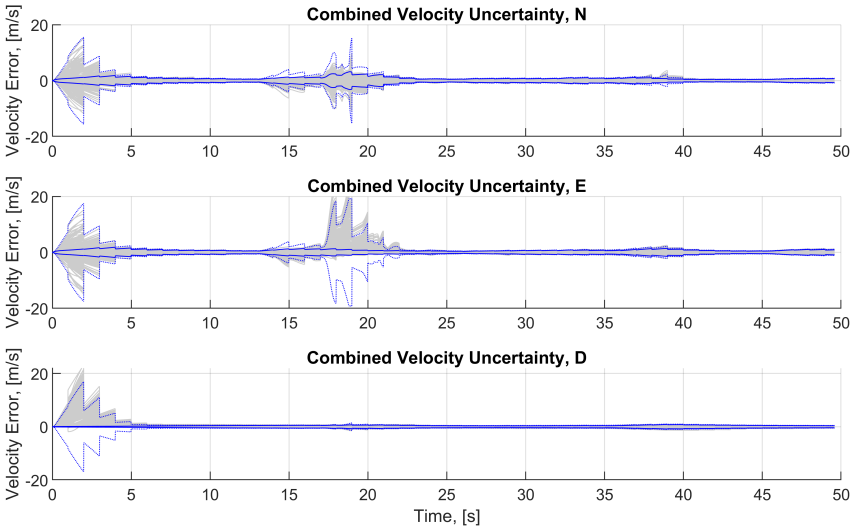


Fig. 5.5: Optimal fusion method results for velocity. Grey lines are 200 Monte Carlo realizations, blue dotted lines are ensemble $3\sigma$, blue solid lines are the predicted $3\sigma$.

the identity constraint in equation 5.13. However, equation 5.3 can now be written as

$$\delta \underline{x}_s = \sum_{i=1}^{n} (\mathbf{I}_3 \otimes \mathbf{\Theta}_i)\mathbf{m}_i \delta \underline{x}_i \tag{5.41}$$

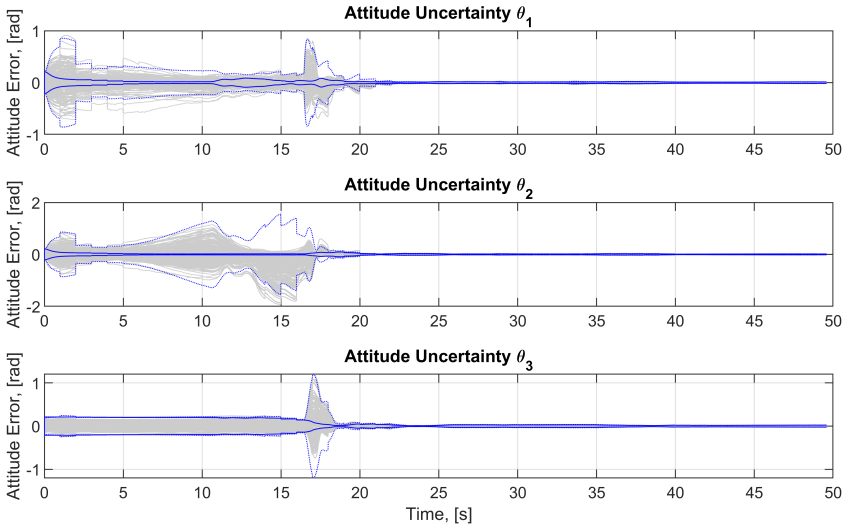


Fig. 5.6: Optimal fusion method results for attitude. Grey lines are 200 Monte Carlo realizations, blue dotted lines are ensemble $3\sigma$, blue solid lines are the predicted $3\sigma$.

This maps the same $\mathbf{\Theta}_i$ to an individual INS's position, velocity, and attitude solutions. The covariance can be expressed as

$$\mathbf{P}_s = \mathrm{E}[\delta \underline{x}_s \delta \underline{x}_s^T] = \sum_{i=1}^{n}\sum_{j=1}^{n} (\mathbf{I}_3 \otimes \mathbf{\Theta}_i)\mathbf{M}_i \mathbf{P}_{15n\times 15n} \mathbf{M}_j^T (\mathbf{I}_3 \otimes \mathbf{\Theta}_j^T) \tag{5.42}$$

Using the same definition of $\mathbf{Q}$ as before, the inner terms can be rewritten,

$$\mathbf{P}_s = \sum_{i=1}^{n}\sum_{j=1}^{n} \begin{bmatrix} \mathbf{\Theta}_i & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{\Theta}_i & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{\Theta}_i \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{ij,11} & \mathbf{Q}_{ij,12} & \mathbf{Q}_{ij,13} \\ \mathbf{Q}_{ij,21} & \mathbf{Q}_{ij,22} & \mathbf{Q}_{ij,23} \\ \mathbf{Q}_{ij,31} & \mathbf{Q}_{ij,32} & \mathbf{Q}_{ij,33} \end{bmatrix} \begin{bmatrix} \mathbf{\Theta}_j^T & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{\Theta}_j^T & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{\Theta}_j^T \end{bmatrix} \tag{5.43}$$

where the $\mathbf{Q}_{ij,kl}$ terms are symmetric block partitions of $\mathbf{Q}$. Carrying out the multiplication leads to

$$\mathbf{P}_s = \sum_{i=1}^{n}\sum_{j=1}^{n} \begin{bmatrix} \mathbf{\Theta}_i\mathbf{Q}_{ij,11}\mathbf{\Theta}_j^T & \mathbf{\Theta}_i\mathbf{Q}_{ij,12}\mathbf{\Theta}_j^T & \mathbf{\Theta}_i\mathbf{Q}_{ij,13}\mathbf{\Theta}_j^T \\ \mathbf{\Theta}_i\mathbf{Q}_{ij,21}\mathbf{\Theta}_j^T & \mathbf{\Theta}_i\mathbf{Q}_{ij,22}\mathbf{\Theta}_j^T & \mathbf{\Theta}_i\mathbf{Q}_{ij,23}\mathbf{\Theta}_j^T \\ \mathbf{\Theta}_i\mathbf{Q}_{ij,31}\mathbf{\Theta}_j^T & \mathbf{\Theta}_i\mathbf{Q}_{ij,32}\mathbf{\Theta}_j^T & \mathbf{\Theta}_i\mathbf{Q}_{ij,33}\mathbf{\Theta}_j^T \end{bmatrix}. \tag{5.44}$$

Since the trace is what will be minimized, it can be applied to equation 5.44 to yield

$$\mathrm{tr}(\mathbf{P}_s) = \sum_{i=1}^{n}\sum_{j=1}^{n}\Big[\mathrm{tr}(\mathbf{\Theta}_i\mathbf{Q}_{ij,11}\mathbf{\Theta}_j^T) + \mathrm{tr}(\mathbf{\Theta}_i\mathbf{Q}_{ij,22}\mathbf{\Theta}_j^T) + \mathrm{tr}(\mathbf{\Theta}_i\mathbf{Q}_{ij,33}\mathbf{\Theta}_j^T)\Big]. \tag{5.45}$$

A simpler notation for the $\mathbf{Q}$ matrix partitions is now adopted,

$$\mathbf{Q}_1 = \mathbf{Q}_{ij,11} \tag{5.46}$$

$$\mathbf{Q}_2 = \mathbf{Q}_{ij,22} \tag{5.47}$$

$$\mathbf{Q}_3 = \mathbf{Q}_{ij,33} \tag{5.48}$$

The problem can now be solved in the same way as in the previous section, resulting in

$$\mathbf{\Theta} = -(\mathbf{Q}_1 + \mathbf{Q}_1^T + \mathbf{Q}_2 + \mathbf{Q}_2^T + \mathbf{Q}_3 + \mathbf{Q}_3^T)^{-1}\sum_{i=1}^{3}\sum_{j=1}^{3}\lambda_{ij}\mathcal{I}^T\underline{e}_i\underline{e}_j^T, \tag{5.49}$$

which is analogous to equation 5.33, and

$$\mathbf{I} = -\sum_{i=1}^{3}\sum_{j=1}^{3}\lambda_{ij}\mathbf{R}\underline{e}_i\underline{e}_j^T \tag{5.50}$$

where

$$\mathbf{R} = \mathcal{I}(\mathbf{Q}_1 + \mathbf{Q}_1^T + \mathbf{Q}_2 + \mathbf{Q}_2^T + \mathbf{Q}_3 + \mathbf{Q}_3^T)^{-1}\mathcal{I}^T. \tag{5.51}$$

Equations 5.50 and 5.51 make up the solution to the optimal constrained fusion method. Noteworthy is that this results in only three equations and three unknowns in equation 5.50, implying that less computation is needed for this solution than for that of section 5.4.

## 5.7 Verification of Optimal Constrained Fusion Method

As with the previous fusion method, the method proposed in section 5.6 was tested with two identical IMUs against the naive average, this time given by

$$\mathbf{\Theta} = 0.5 \begin{bmatrix} \mathbf{I}_{3\times 3} \\ \mathbf{I}_{3\times 3} \end{bmatrix}. \tag{5.52}$$

It should be noted that this naive average produces precisely the same result as equation 5.36 in this framework, so the reader can refer to section 5.5 for those results. The constrained optimal fusion method performed comparatively better than the unconstrained optimal fusion method. Figs. 5.7-5.9 show the results when there is no correlation in the $15n \times 15n$ covariance matrix, and it results in good agreement between the predicted variances and ensemble averages. The down velocity component has some noted abnormalities, but the majority of the solutions outperform the naive average.

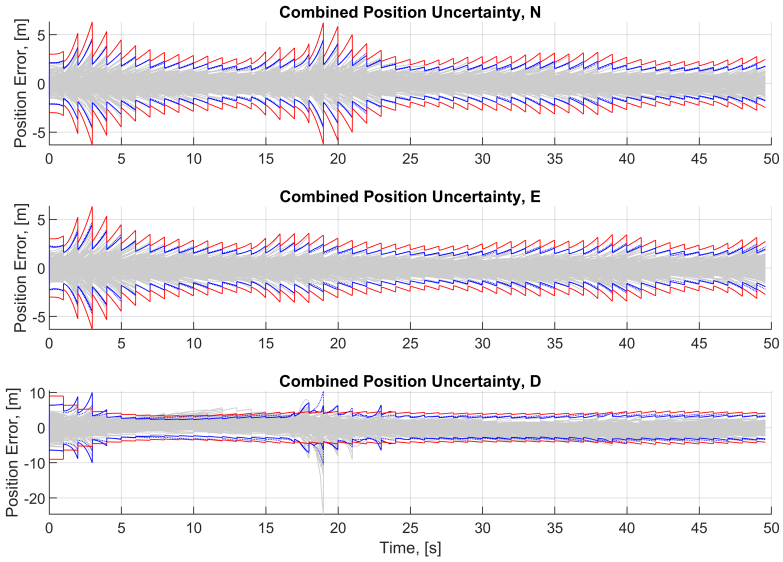


Fig. 5.7: Optimal constrained fusion method results for position with uncorrelated master covariance matrix. Grey lines are 200 Monte Carlo realizations, blue dotted lines are ensemble $3\sigma$, blue solid lines are the predicted $3\sigma$, and red solid lines are the single IMU $3\sigma$ for comparison.

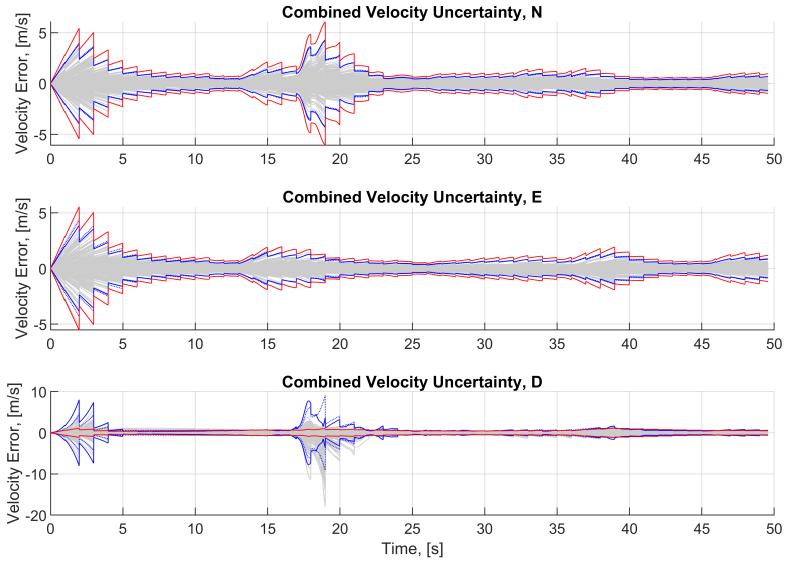


Fig. 5.8: Optimal constrained fusion method results for velocity with uncorrelated master covariance matrix. Grey lines are 200 Monte Carlo realizations, blue dotted lines are ensemble $3\sigma$, blue solid lines are the predicted $3\sigma$, and red solid lines are the single IMU $3\sigma$ for comparison.

The uncorrelated results are almost identical to the naive fusion method as seen in Figs. 5.1-5.3. This is because IMUs with the same specs and same lever arms were used, so a naive weighted average is identical to the solution of the constrained optimal fusion method. However, in the case of varying specifications, uncertain lever arms, and complicated geometries, the naive average will be difficult to estimate. The optimal constrained algorithm was derived to handle these factors and weight the outputs of the INSs dynamically.

When a GPS measurement is shared between the individual INSs and correlations are introduced into the covariance matrix, the algorithm still performs sufficiently, as is shown in Figs. 5.10-5.12. The predicted $3\sigma$ variances of all three quantities do not match the ensemble average $3\sigma$ variances perfectly but they still represent a marked improvement over the single IMU case shown in Figs. 3.2-3.4, with the exception of the down-direction velocity. These results merit further investigation and generalization to more IMUs and different geometries in Chapter 6.

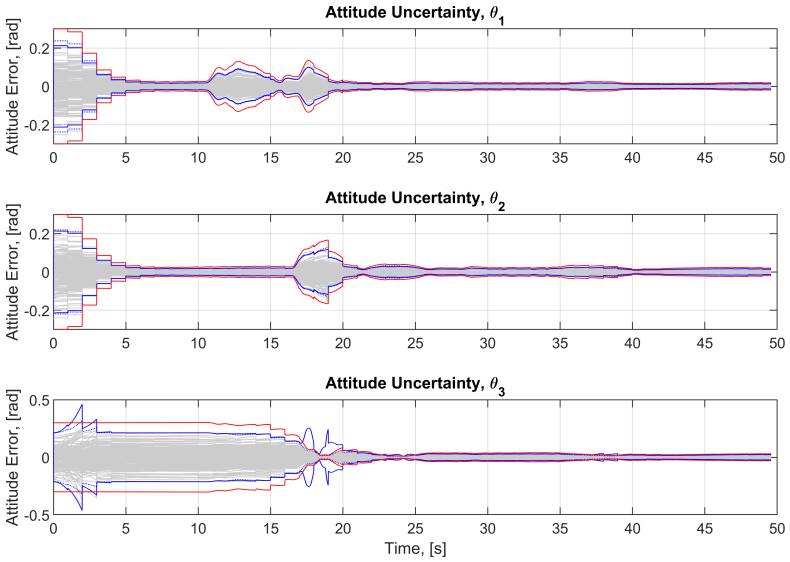


Fig. 5.9: Optimal constrained fusion method results for attitude with uncorrelated master covariance matrix. Grey lines are 200 Monte Carlo realizations, blue dotted lines are ensemble $3\sigma$, blue solid lines are the predicted $3\sigma$, and red solid lines are the single IMU $3\sigma$ for comparison.

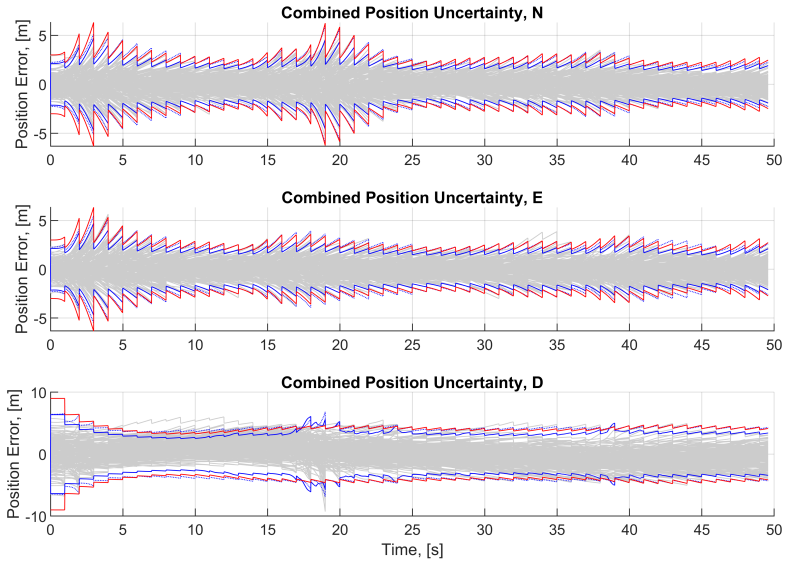


Fig. 5.10: Optimal constrained fusion method results for position with correlated master covariance matrix. Grey lines are 200 Monte Carlo realizations, blue dotted lines are ensemble $3\sigma$, blue solid lines are the predicted $3\sigma$, and red solid lines are the single IMU $3\sigma$ for comparison.

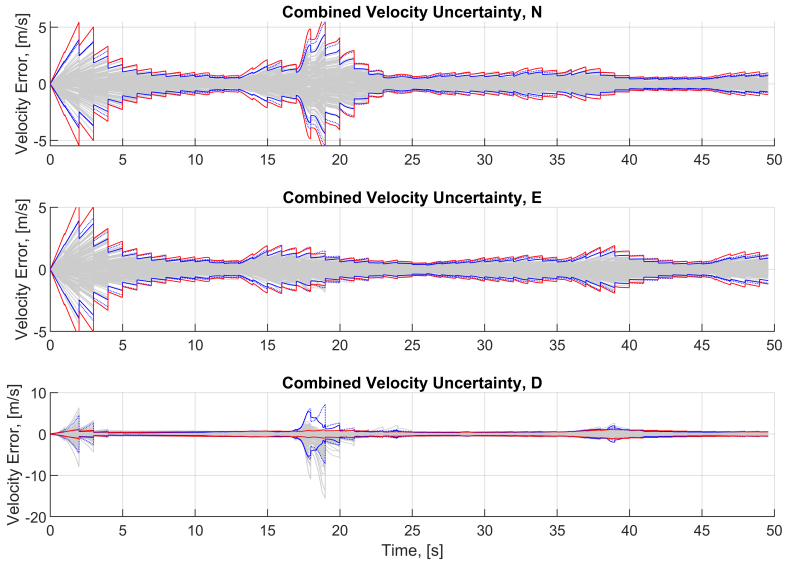


Fig. 5.11: Optimal constrained fusion method results for velocity with correlated master covariance matrix. Grey lines are 200 Monte Carlo realizations, blue dotted lines are ensemble $3\sigma$, blue solid lines are the predicted $3\sigma$, and red solid lines are the single IMU $3\sigma$ for comparison.

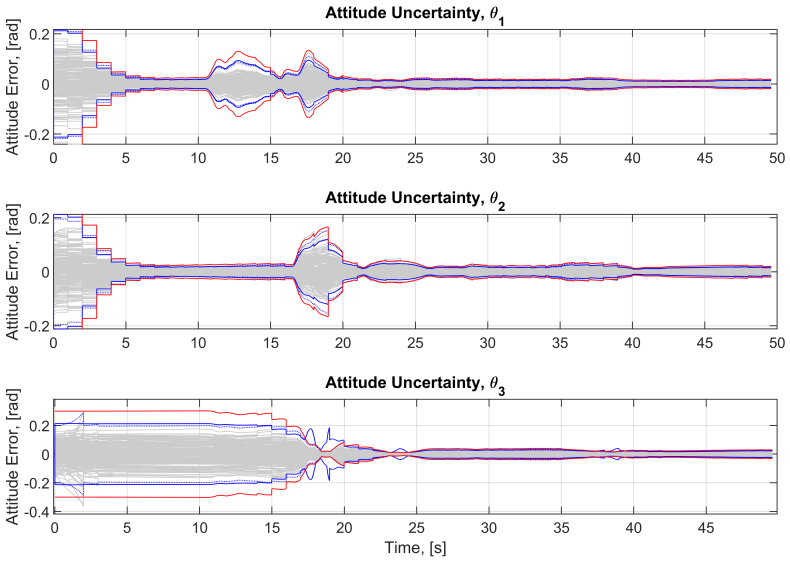


Fig. 5.12: Optimal constrained fusion method results for attitude with correlated master covariance matrix. Grey lines are 200 Monte Carlo realizations, blue dotted lines are ensemble $3\sigma$, blue solid lines are the predicted $3\sigma$, and red solid lines are the single IMU $3\sigma$ for comparison.

# CHAPTER 6

# PERFORMANCE OF THE OPTIMAL CONSTRAINED FUSION ALGORITHM

In the previous chapter, the optimal constrained fusion algorithm (OCFA) was derived and shown to match the performance of a naive equal weighted average in a simple case. Additionally, the algorithm has much more capability and modularity than an uninformed average. Taking advantage of all of the information in the master covariance matrix, the OCFA can automatically weight the better state estimates and synthesize a better solution with arbitrary geometric and specification configurations.

Furthermore, the ability to utilize the correlations caused by repeated measurements and to produce reasonable results is made more valuable by the fact that multiple GPS observations are not independent. Most GPS error sources will be shared between multiple antennae, including multipath, ionospheric, tropospheric, and satellite positioning errors. This means that the measurements would not be 100% correlated but they would also not be totally independent, implying that the two cases being tested in this work, "correlated" and "uncorrelated," are both unrealistic in the multiple-GPS scenario and represent the worst and best cases respectively. The "correlated" category would be realistic for a multi-INS system that shared the measurements of a single GPS antenna.

This chapter will present many simulation results that were generated using the algorithm in order to understand its performance and usefulness in a broader context. For each simulation case, the mean of the $3\sigma$ standard deviation of the errors of each component of the solution over all times was recorded as a performance metric. The true trajectory is the same each time and the only variables are the numbers of IMUs and the lengths and distributions of the lever arms.

## 6.1 Improving Performance Iteratively

As mentioned previously, one potential improvement in the solution could come from

Table 6.1: Results of Iterative OCFA, 2-IMU, correlated measurements

| Description | 2-IMU | 2-IMU, 2nd iteration | Units | % change |
|---|---|---|---|---|
| Mean of ensemble $3\sigma$ position, N | 2.5740 | 2.5734 | m | -0.02 |
| Mean of ensemble $3\sigma$ position, E | 2.5612 | 2.5616 | m | 0.02 |
| Mean of ensemble $3\sigma$ position, D | 4.3563 | 4.3566 | m | 0.01 |
| Mean of covariance matrix $3\sigma$ position, N | 2.0510 | 2.0515 | m | 0.02 |
| Mean of covariance matrix $3\sigma$ position, E | 1.9855 | 1.9857 | m | 0.01 |
| Mean of covariance matrix $3\sigma$ position, D | 3.6332 | 3.6335 | m | 0.01 |
| Mean of ensemble $3\sigma$ velocity, N | 1.2002 | 1.2004 | m/s | 0.02 |
| Mean of ensemble $3\sigma$ velocity, E | 1.0779 | 1.0778 | m/s | 0.00 |
| Mean of ensemble $3\sigma$ velocity, D | 0.8684 | 0.8674 | m/s | -0.11 |
| Mean of covariance matrix $3\sigma$ velocity, N | 1.0312 | 1.0312 | m/s | 0.00 |
| Mean of covariance matrix $3\sigma$ velocity, E | 0.8941 | 0.8941 | m/s | 0.00 |
| Mean of covariance matrix $3\sigma$ velocity, D | 0.8090 | 0.8092 | m/s | 0.02 |
| Mean of ensemble $3\sigma$ attitude, $\theta_1$ | 0.0360 | 0.0360 | rad | 0.00 |
| Mean of ensemble $3\sigma$ attitude, $\theta_2$ | 0.0383 | 0.0383 | rad | 0.00 |
| Mean of ensemble $3\sigma$ attitude, $\theta_3$ | 0.0915 | 0.0915 | rad | 0.00 |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_1$ | 0.0339 | 0.0339 | rad | 0.00 |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_2$ | 0.0340 | 0.0340 | rad | 0.00 |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_3$ | 0.0932 | 0.0932 | rad | 0.00 |

running the algorithm again using the improved estimate of $q^b_{ned}$. In equation 5.4, the derivation includes the rotation matrix $\mathbf{R}^{ned}_{\hat{b}}$, which was calculated from each IMUs attitude solution. Because the algorithm generates a combined attitude solution that has a lower standard deviation than an individual solution, it is possible to recalculate the rotation matrix and run the algorithm a second time, this time with the new, more accurate rotation matrix.

A comparison of the combined solution and the iterative combined solution is shown in Table 6.1. The ensemble results were generated with 250 Monte Carlo runs. The two columns of data are equivalent down to the hundredths of a percent, likely implying that running the algorithm a second time does not offer any noticeable advantages.

Table 6.2: MEMS and Tactical IMU Parameters

| | Description | Symbol | Value | Units |
|---|---|---|---|---|
| MEMS | Accelerometer ECRV time constant | $\tau_{a,1}$ | 60 | s |
| | Gyroscope ECRV time constant | $\tau_{g,1}$ | 60 | s |
| | Steady state accelerometer bias $1\sigma$ | $\sigma_{ss,a,1}$ | 0.019613 | $m/s^2$ |
| | Steady state gyroscope bias $1\sigma$ | $\sigma_{ss,g,1}$ | 0.001212 | rad/s |
| | Velocity random walk | $\text{vrw}_1$ | 0.001177 | $m/s^2/\sqrt{Hz}$ |
| | Angular random walk | $\text{arw}_1$ | $4.3633 \times 10^{-4}$ | $rad/s/\sqrt{Hz}$ |
| | Description | Symbol | Value | Units |
| Tactical | Accelerometer ECRV time constant | $\tau_{a,2}$ | 360 | s |
| | Gyroscope ECRV time constant | $\tau_{g,2}$ | 360 | s |
| | Steady state accelerometer bias $1\sigma$ | $\sigma_{ss,a,2}$ | 0.001961 | $m/s^2$ |
| | Steady state gyroscope bias $1\sigma$ | $\sigma_{ss,g,2}$ | 0.000121 | rad/s |
| | Velocity random walk | $\text{vrw}_2$ | $1.5691 \times 10^{-4}$ | $m/s^2/\sqrt{Hz}$ |
| | Angular random walk | $\text{arw}_2$ | $3.4907 \times 10^{-5}$ | $rad/s/\sqrt{Hz}$ |

## 6.2 Mixing IMU Specifications

In the scenario that multiple IMUs are available but have different specifications, the OCFA can be employed to weight their INS solutions dynamically. This was tested by specifying a first IMU as having specifications similar to those of previous sections and a second with tactical-level specifications. The tactical IMU is an Analog Devices ADIS16490. The lever arms were both made equal to one meter and the GPS measurements were shared. Table 6.2 shows the specifications used in this study for each IMU. All other simulation values are equal to those given in Table 3.1.

Elements of the weighting matrix, $\mathbf{\Theta}$, were then recorded over time. It was expected that the algorithm would favor the IMU with higher quality specifications and weight its solutions accordingly based on the information present in the master covariance matrix. Figs. 6.1-6.2 show the weighting components over time for 2-IMUs and 4-IMUs.

The results show that the algorithm favors the IMU with the better specifications as predicted. In addition, the sum of the diagonal components in the weighting matrix equals one, confirming that it is following the constraint posed in Chapter 5.

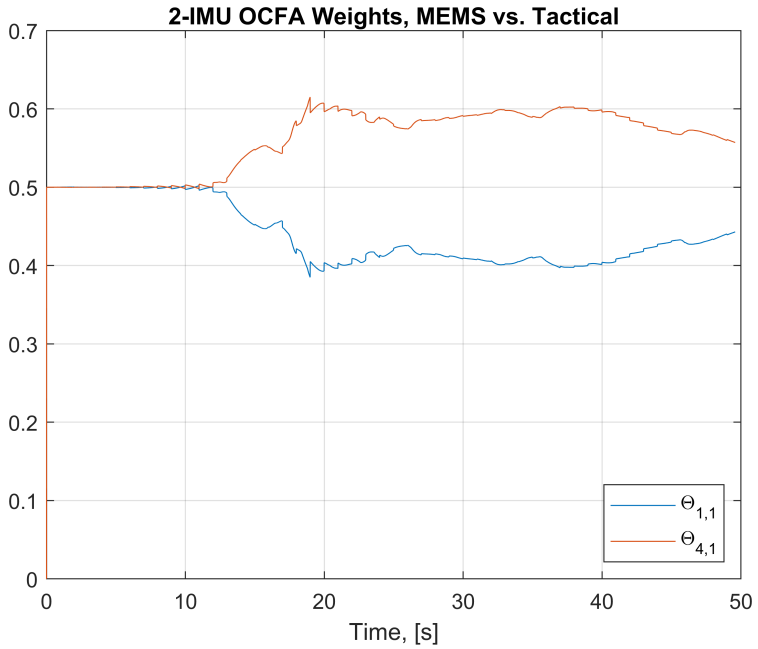


Fig. 6.1: OCFA weights calculated over time for IMUs of differing specifications. The second IMU is clearly weighted more than the first.

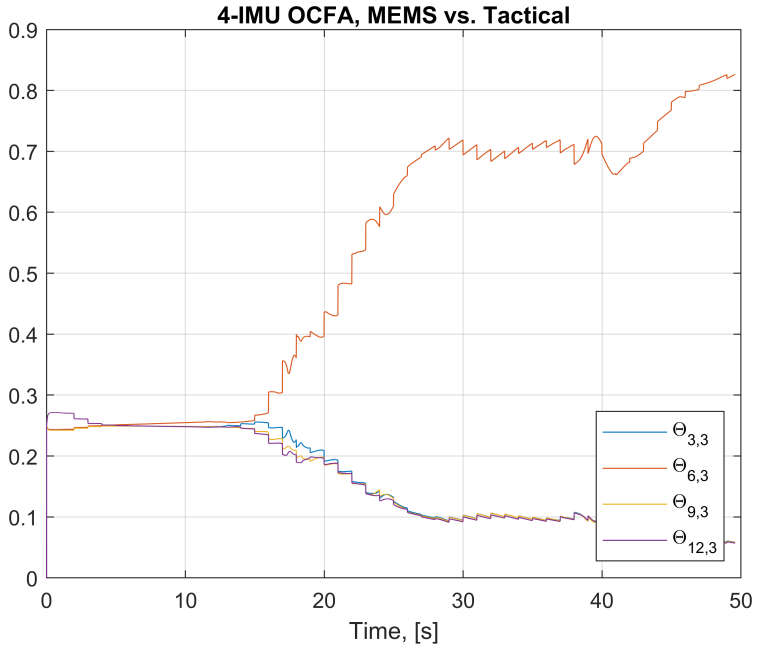


Fig. 6.2: OCFA weights calculated over time for IMUs of differing specifications. The second IMU is clearly weighted more than the first, third, and fourth.

### 6.3 Geometry Study

It is difficult to predict how the solutions will behave as the number of IMUs and their respective lever arms increase. Increasing the number of IMUs, and thereby increasing the amount of information available, was expected to improve the solution. Additionally, because the accelerometers are not being leveraged in the calculation of the angular rates, as was the case in the observation domain strategies in Chapter 4, it was anticipated that performance would actually suffer with larger lever arms due to errors in the relative attitude between sensors. The result would therefore be optimal lever arms of zero.

In order to test these hypotheses, a testing battery was designed and executed based on the number of IMUs and various configuration geometries. Each test generated 200 Monte Carlo runs which were used to generate the ensemble means. The basic shapes used in the tests are shown in Fig. 6.3. The IMUs were assumed to be aligned and to have perfectly-known lever arms. For the case of three IMUs, there were two configurations tested: one where each IMU is at a lever arm from the central location of combination and one where the third IMU is placed at said point. These two 3-IMU configurations are equal in the degenerate case when the lever arms equal zero. The other configurations were all symmetric, with the 5-IMU and 7-IMU cases having the last IMU at the central location. Each IMU is assumed to have a GPS antenna at its location. In the correlated case, the same noise is fed to each GPS antenna.

Each of these configurations was simulated for both correlated and uncorrelated cases for lever arms of zero meters, one meter, and three meters. The data generated is tabulated in Tables D.1-D.6 in Appendix D, which is shown as plots in Figs. 6.4-6.9 here. In the uncorrelated cases, the trends in the figures show that all of the multi-IMU configurations performed appreciably better than the single IMU control in all three combined states. In the uncorrelated cases, the 2-IMU, 4-IMU, 5-IMU, 6-IMU, and 7-IMU cases produced better attitude solutions while offering little-to-no benefit in the position solutions. The velocity solutions were also improved with the exception of the down direction.

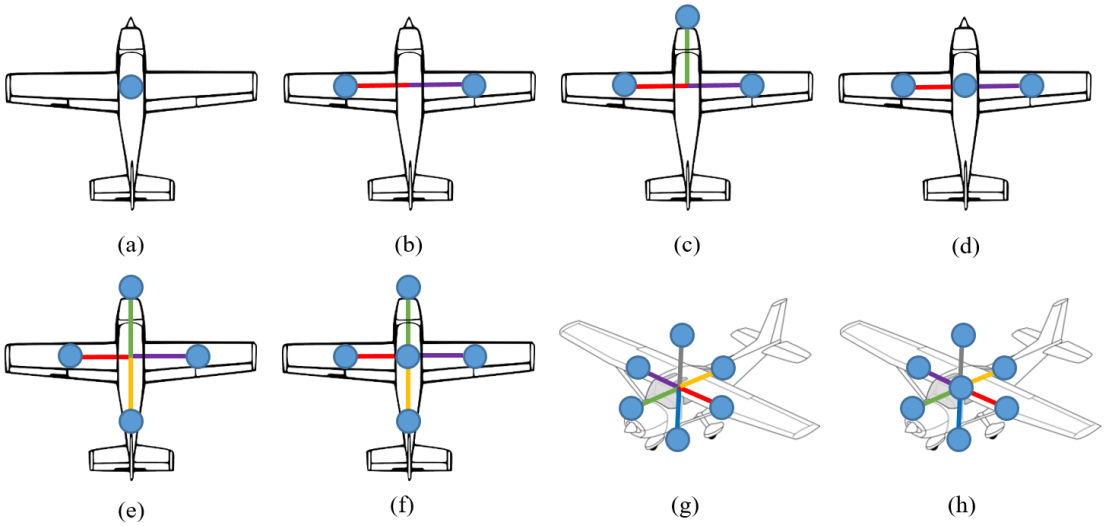


Fig. 6.3: Tested IMU configurations with IMUs denoted by blue circles and arbitrary-length lever arms denoted by various colors: (a) 1 IMU, (b) 2 IMUs, (c) 3 IMUs in a triangular shape, (d) 3 IMUs in line, (e) 4 IMUs, (f) 5 IMUs, (g) 6 IMUs, (h) 7 IMUs.

On the other hand, the 3-IMU configurations did not follow the general trends, especially in the correlated cases. The reason for this is not completely clear. Further investigation reveals that for the 3-IMU configurations, the condition numbers of the matrix partitions $\mathbf{Q}_{ii}$, defined in equation 5.46, spike up to high values at certain points in the trajectory. This happens to a lesser extent in the other configurations.

The effect of geometry on the solution is as expected: larger lever arms result in larger standard deviations, although the effect is small in most cases. Plots of these results for the cases with correlated measurements can be seen in Figs. 6.10-6.12, where only the first component of each state is shown. These trends are also representative of the east and down components, with the exception of the down velocity component. A plot of the effect of geometry on the down velocity component, which is anomalous and does not follow the trends of the other components, is shown in Fig. 6.13 for comparison.

Returning to the more promising cases, it is noted that, in general, fusion with correlated measurements does not perform as well as fusion with uncorrelated measurements. Where the position and velocity solutions are on the order of 30% to 60% better than the

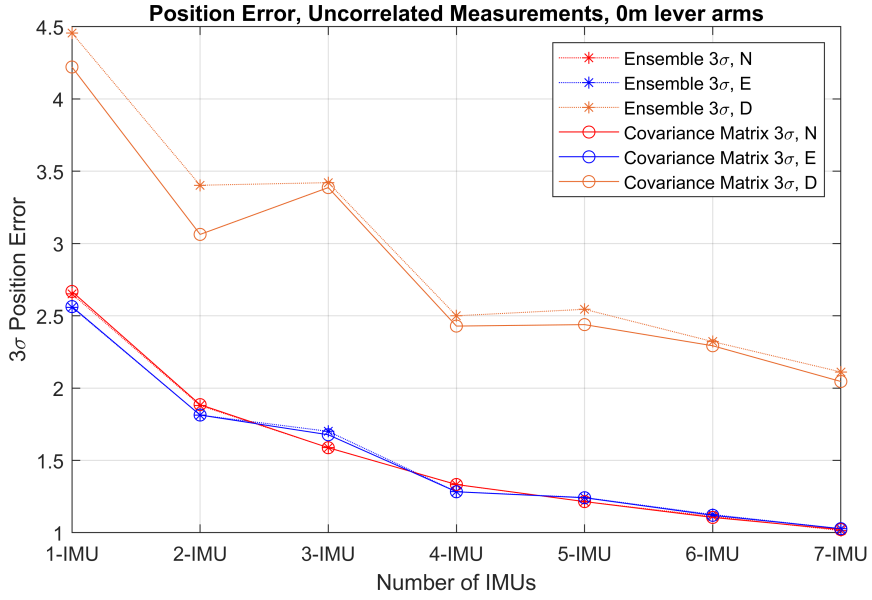


Fig. 6.4: Geometry Study: position error with uncorrelated measurements vs. number of IMUs, 0m lever arms.

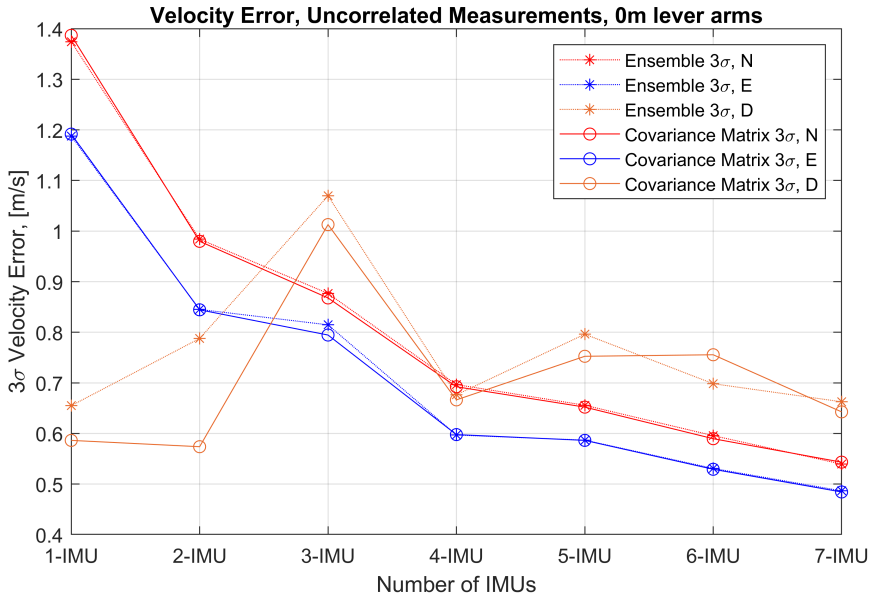


Fig. 6.5: Geometry Study: velocity error with uncorrelated measurements vs. number of IMUs, 0m lever arms.

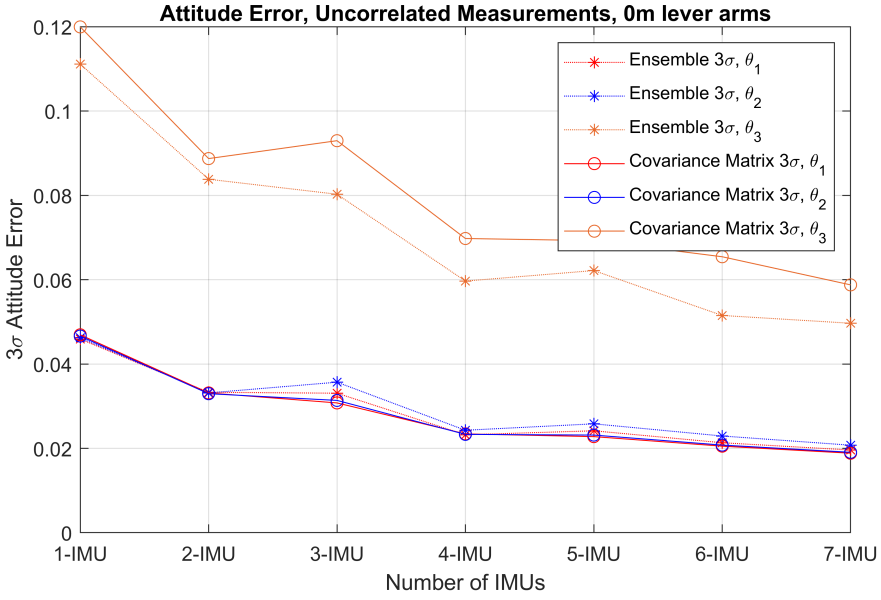


Fig. 6.6: Geometry Study: attitude error with uncorrelated measurements vs. number of IMUs, 0m lever arms.

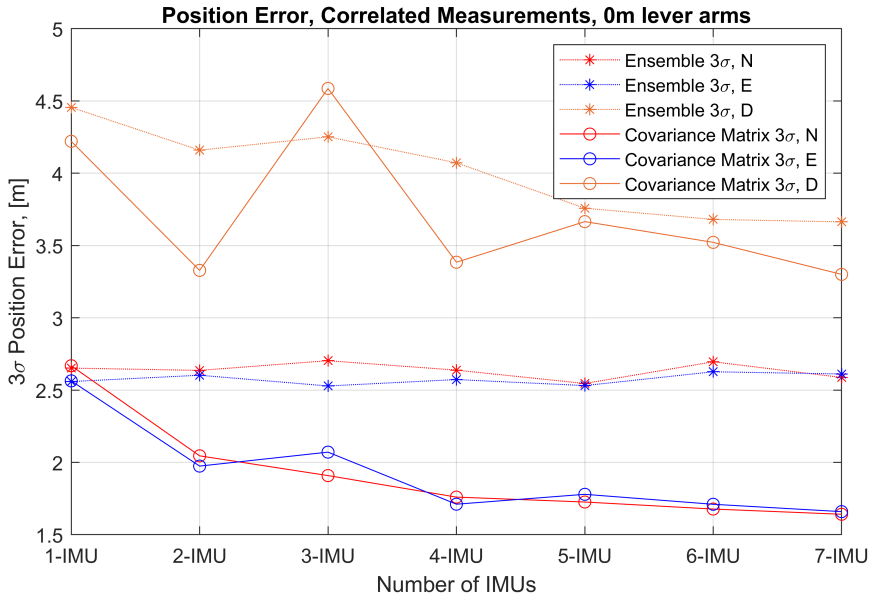


Fig. 6.7: Geometry Study: position error with correlated measurements vs. number of IMUs, 0m lever arms.

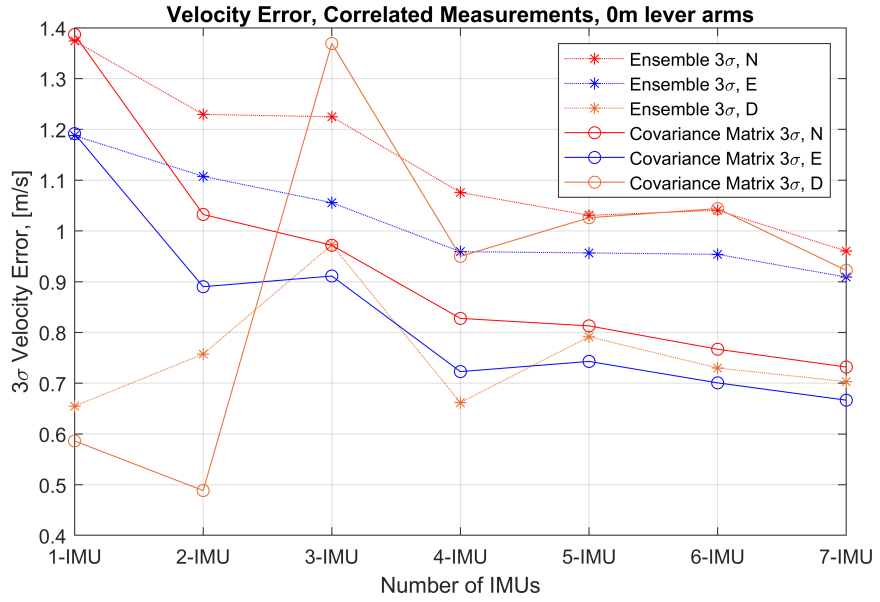


Fig. 6.8: Geometry Study: velocity error with correlated measurements vs. number of IMUs, 0m lever arms.

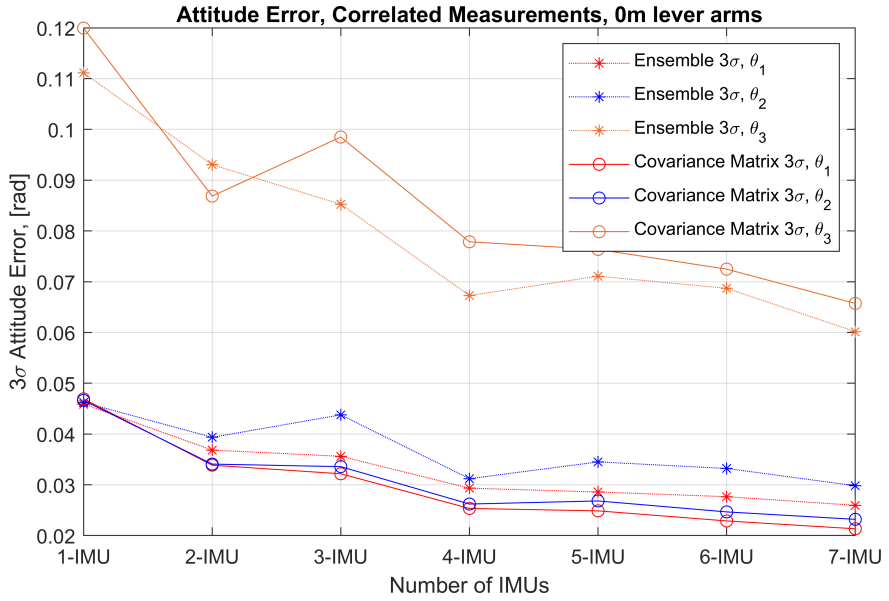


Fig. 6.9: Geometry Study: attitude error with uncorrelated measurements vs. number of IMUs, 0m lever arms.

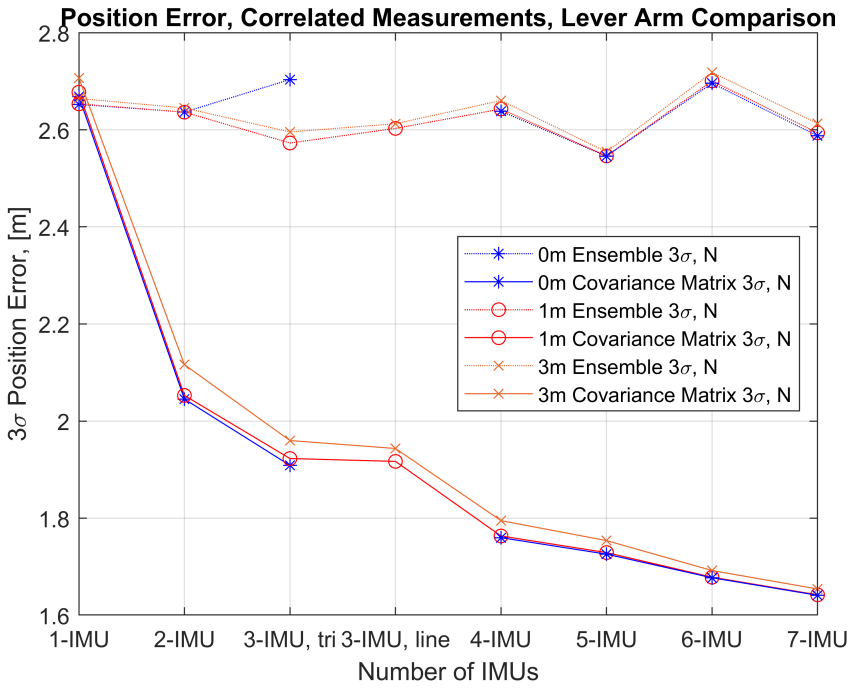


Fig. 6.10: Geometry Study: lever arm length effect on north position solution.

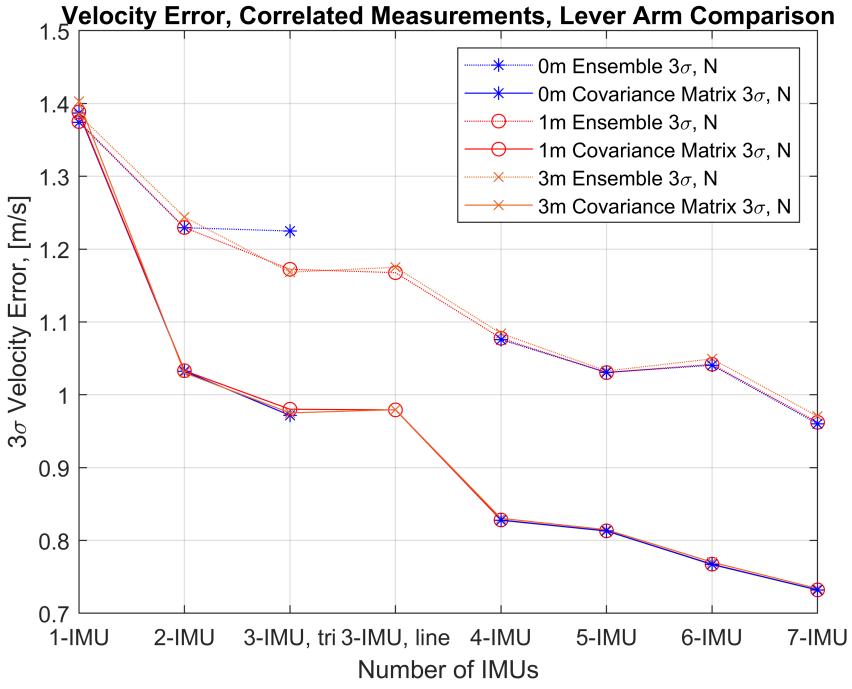


Fig. 6.11: Geometry Study: lever arm length effect on north velocity solution.

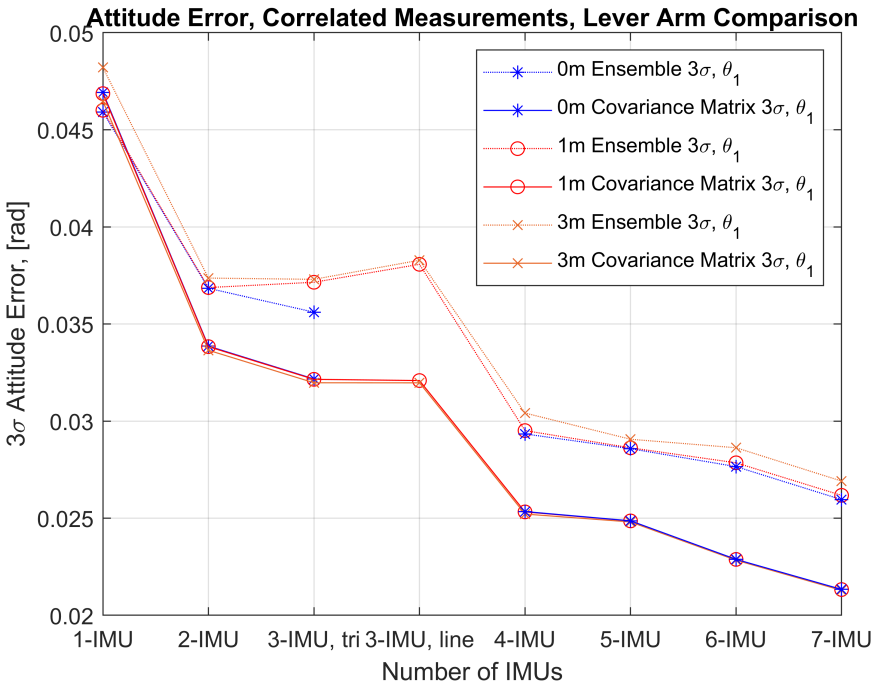


Fig. 6.12: Geometry Study: lever arm length effect on $\theta_1$ attitude solution.

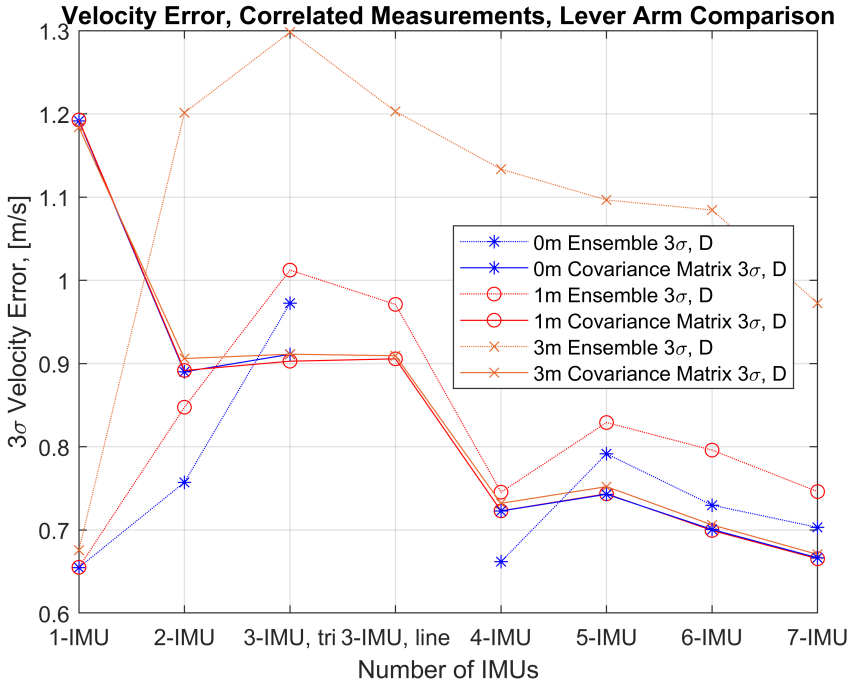


Fig. 6.13: Geometry Study: lever arm length effect on down velocity solution.

Table 6.3: 7-IMU Simulation Results, 0 meter lever arms

| Description | Uncorrelated | % change | Correlated | % change |
|---|---|---|---|---|
| Mean of ensemble $3\sigma$ position, N | 1.0146 m | -61.75 | 2.5874 m | -2.44 |
| Mean of ensemble $3\sigma$ position, E | 1.0276 m | -59.85 | 2.6116 m | 2.05 |
| Mean of ensemble $3\sigma$ position, D | 2.1116 m | -52.61 | 3.6636 m | -17.78 |
| Mean of covariance matrix $3\sigma$ position, N | 1.0212 m | -61.73 | 1.6412 m | -38.49 |
| Mean of covariance matrix $3\sigma$ position, E | 1.0274 m | -59.92 | 1.6594 m | -35.28 |
| Mean of covariance matrix $3\sigma$ position, D | 2.0457 m | -51.53 | 3.3004 m | -21.81 |
| Mean of ensemble $3\sigma$ velocity, N | 0.5395 m/s | -60.75 | 0.9602 m/s | -30.13 |
| Mean of ensemble $3\sigma$ velocity, E | 0.4866 m/s | -59.03 | 0.9091 m/s | -23.46 |
| Mean of ensemble $3\sigma$ velocity, D | 0.6626 m/s | 1.17 | 0.7031 m/s | 7.35 |
| Mean of covariance matrix $3\sigma$ velocity, N | 0.5433 m/s | -60.82 | 0.7319 m/s | -47.22 |
| Mean of covariance matrix $3\sigma$ velocity, E | 0.4844 m/s | -59.35 | 0.6666 m/s | -44.06 |
| Mean of covariance matrix $3\sigma$ velocity, D | 0.6427 m/s | 9.61 | 0.9222 m/s | 57.30 |
| Mean of ensemble $3\sigma$ attitude, $\theta_1$ | 0.0196 rad | -57.29 | 0.0260 rad | -43.47 |
| Mean of ensemble $3\sigma$ attitude, $\theta_2$ | 0.0207 rad | -55.14 | 0.0298 rad | -35.53 |
| Mean of ensemble $3\sigma$ attitude, $\theta_3$ | 0.0497 rad | -55.30 | 0.0602 rad | -45.86 |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_1$ | 0.0188 rad | -59.84 | 0.0213 rad | -54.54 |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_2$ | 0.0191 rad | -59.11 | 0.0232 rad | -50.24 |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_3$ | 0.0588 rad | -51.01 | 0.0657 rad | -45.21 |

control case, the correlated solutions of these quantities tended to be only 5% to 45% improved, if at all. The exception seems to be the attitude solutions, which tend to perform 35% to 40% better whether the measurements were correlated or not.

## 6.4 Performance During Flight

The seven-IMU, zero-lever arm configuration seems to perform the best overall among the tested configurations. The errors of this case are shown in Table 6.3. The implementation of this configuration could be done on a single printed circuit board and yield attitude solutions that have standard deviations that are, on average, 41% lower than a single MEMS IMU.

Furthermore, when the aircraft experiences motion starting around 15 seconds, the attitude errors are drastically reduced. This regime is likely a better representation of the magnitude of errors that would be recorded in a real sUAS photogrammetry flight because the aircraft would be maneuvering frequently and the amount of information for the algorithm would be higher. The roll and pitch attitude errors at $t = 30$ seconds have a $3\sigma$ of approximately $0.8^{\circ}$ and the heading error is reduced to a $3\sigma$ of $1.85^{\circ}$. While not

matching the $0.1^{o}$ and $0.3^{o}$ of the VectorNav VN-200 mentioned in Chapter 1, this is a marked improvement over the single MEMS IMU case. For comparison the single MEMS IMU produces an attitude solution at $t = 30$ seconds with a $3\sigma$ of $2.68^{o}$ in pitch and roll and $6.9^{o}$ in heading. It should be noted that this improvement does come at the cost of increased computational demand, but this is likely an issue that can be overcome.

# CHAPTER 7

# CONCLUSION AND FUTURE WORK

This research project is both an investigation of prior methods of multi-IMU sensor fusion and attempts to add to the existing body of knowledge of the subject. This problem has been approached in many ways and most of the existing work fell within the observation domain category. The estimation domain category remains rather underexplored despite its advantages, and this work has presented a novel approach to solving a problem that was considered a dead end in prior work.

The question of whether or not existing observation domain strategies would be useful in an sUAS/MEMS context was also investigated. It is shown that the state-of-the-art gyroscope free algorithm is still insufficient to provide good angular velocity data to a navigation system using the best-available MEMS accelerometers at the time of writing. The maximum likelihood estimator, proposed more recently, was shown to have only marginal benefits for this application in its current form. The possible benefits from successful implementation remains due to the extra information and capability that it provides, especially at high angular rates.

Two estimation domain fusion strategies were derived and tested. The unconstrained version, which estimated a weighting matrix $\mathbf{\Theta}_{9n\times 9}$, resulted in anomalous results and poor performance. This motivated the derivation of a similar algorithm that was instead constrained in such a way as to calculate only a $\mathbf{\Theta}_{3n\times 3}$ weighting matrix, applying the same weights to the position, velocity, and attitude solutions for each respective INS. The performance of this algorithm was much better, and extensive simulation and testing was done and presented in Chapter 6. Though the unconstrained approach showed generally improved performance, there were still some questionable results. The predicted variances in Figs. 5.7-5.12 do not match the ensemble statistics very well, especially in the cases with correlated measurements. The reasons for this are not known at the time of writing and

merit further investigation. It is noteworthy, however, that the combined covariance matrix that produced these variance plots is not used in any further calculations, as would be the case in a standard Kalman filter.

It was shown that increasing the number of IMUs used in the OCFA generally increases the fidelity of the solution. The algorithm was shown to be suitable for configurations of two, four, five, six, and seven IMUs. Cases involving three IMUs resulted in an ill-conditioned matrix inversion and poor performance. The solution produces less accurate results as the lever arms to the IMUs are increased, which may actually prove to be beneficial in a real-time system. The algorithm is also capable of handling IMUs of varying specifications, automatically weighting a tactical-grade IMU over a MEMS IMU.

## 7.1 Future Work

In the future, the optimal constrained fusion algorithm could be validated by implementation and testing on a real platform. This would involve the evaluation of both computational and timing constraints prior to real-time hardware integration. If the algorithm is determined to be insufficient for real-time applications, it could still find uses in data post processing. Further evaluation using larger numbers of IMUs could also be done.

The ability to handle correlations introduced by shared measurement information may have other applications and it is anticipated that future work will leverage the OCFA for this purpose. One such application could be system identification: the estimation of characteristics of the system such as lever arms and bending modes. Another could be leveraging the correlations in the master covariance matrix for attitude estimation using multiple GNSS antennae.

It is also worth noting that, in its current form, the algorithm cannot do any kind of fault detection and isolation (FDI), an important possible benefit of multi-IMU systems. That said, an FDI algorithm could be worked into the previously shown formulation in various ways, whether as a supplementary, high-level algorithm or embedded as another constraint of the algorithm itself.

## 7.2 Lessons Learned

A problem needs to be stated well in order to be answered sufficiently. Among the direct knowledge of the subjects at hand in this project, I learned that being organized, writing things down, and coming up with a cohesive plan before even attempting to solve a complex problem is crucial. If this advice is not heeded, it can be easy to become overwhelmed with the sheer amount of information published in this field.

A large amount of time was spent on the literature review section of this thesis and I hope that future researchers will be able to use this work as a basis for their projects. Much of the literature is very difficult for a first-year graduate student to approach, so I have tried to identify what I perceived as the most useful strategies for this particular application and how they fit into a useful classification structure. Some background in inertial navigation, control theory, estimation theory, and optimization theory is very helpful before attempting to engage with the professional literature on this topic.

# REFERENCES


[1] P. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2013.

[2] "Inertial Measurement Units and Inertial Navigation." [Online]. Available: https://www.vectornav.com/support/library/imu-and-ins

[3] P. Maybeck, *Stochastic Models, Estimation and Control: Volume I*. Academic Press, 1979.

[4] "AggieAir Flying Circus." [Online]. Available: https://uwrl.usu.edu/aggieair/

[5] H. Chao, C. Coopmans, L. Di, and Y. Chen, "A Comparative Evaluation of Low Cost IMUs for Unmanned Autonomous Systems," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2010, pp. 211–216.

[6] M. Givens and C. Coopmans, "A Survey of Inertial Sensor Fusion: Applications in sUAS Navigation and Data Collection," in *2019 International Conference on Unmanned Aerial Systems*. Atlanta, GA: IEEE, 2019.

[7] J. B. Bancroft and G. Lachapelle, "Data Fusion Algorithms for Multiple Inertial Measurement Units," *Sensors*, vol. 11, no. 12, pp. 6771–6798, 6 2011.

[8] I. Skog, J.-O. Nilsson, P. Handel, and A. Nehorai, "Inertial Sensor Arrays, Maximum Likelihood, and CramérRao Bound," *IEEE Transactions on Signal Processing*, vol. 64, no. 16, pp. 4218–4227, 8 2016.

[9] I. Colomina, M. Gimenez, J.J. Rosales, M. Wis, A. Gomez, and P. Miguelsanz, "Redundant IMUs for Precise Trajectory Determination," *XXth ISPRS Congress*, p. P.7, 2004.

[10] R. Hanson and M. Pachter, "Optimal Gyro-Free IMU Geometry," in *AIAA Guidance, Navigation, and Control Conference*, 2005.

[11] V. Krishnan, "Measurement of angular velocity and linear acceleration using linear accelerometers," *Journal of the Franklin Institute*, vol. 280, no. 4, pp. 307–315, 10 1965.

[12] A. Schuler, Alfred R. Grammatikos, "Measuring Rotational Motion with Linear Accelerometers," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-3, no. 3, pp. 465 – 472, 1967.

[13] J.-O. Nilsson and I. Skog, "Inertial Sensor Arrays  A Literature Review," in *2016 European Navigation Conference (ENC)*, 2016.

[14] K. Parsa, J. Angeles, and A. K. Misra, "Rigid-body pose and twist estimation using an accelerometer array," *Archive of Applied Mechanics*, vol. 74, no. 3-4, pp. 223–236, 2004.

[15] P. Cardou and J. Angeles, "A Comparative Study of All-Accelerometer Strapdowns for UAV INS," in *Meeting Proceedings RTO-MP-SET-092*, 2005, pp. 9–1.

[16] M. Pachter, T. C. Welker, and R. E. Huffman, "Gyro-free INS theory," *Journal of the Institute of Navigation*, vol. 60, no. 2, pp. 85–96, 2013.

[17] M. Rhudy, J. Gross, Y. Gu, and M. Napolitano, "Fusion of GPS and Redundant IMU Data for Attitude Estimation," in *AIAA Guidance, Navigation, and Control Conference*, no. May 2014, 2012.

[18] A. J. Pesja, "Optimum Skewed Redundant Inertial Navigators," *AIAA Journal*, vol. 12, no. 7, pp. 899–902, 7 1974.

[19] M. Sturza, "Skewed axis inertial sensor geometry for optimal performance," in *Digital Avionics Systems Conference*. Reston, Virigina: American Institute of Aeronautics and Astronautics, 10 1988.

[20] J. Guerrier, Stéphane; Skaloud, "Integration of Skew-Redundant MEMS-IMU with GPS for Improved Navigation Performance," Ph.D. dissertation, 2008.

[21] R. Giroux, "Orthogonal vs Skewed Inertial Sensors Redundancy: A New Paradigm for Low-Cost Systems," in *IFAC Proceedings Volumes*, vol. 37, no. 5, 2004, pp. 97–102.

[22] A. Waegli, S. Guerrier, and J. Skaloud, "Redundant MEMS-IMU integrated with GPS for performance assessment in sports," in *Record - IEEE PLANS, Position Location and Navigation Symposium*, 2008, pp. 1260–1268.

[23] S. Guerrier, "Improving Accuracy with Multiple Sensors : Study of Redundant MEMS-IMU / GPS Configurations," in *Proceedings of the 22nd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2009)*, no. May, 2009, pp. 3114–3121.

[24] D. S. Bayard and S. R. Ploen, "High accuracy inertial sensors from inexpensive components," 3 2003.

[25] H. Chang, L. Xue, W. Qin, G. Yuan, and W. Yuan, "An Integrated MEMS Gyroscope Array with Higher Accuracy Output," *Sensors*, vol. 8, pp. 2886–2899, 2008.

[26] J. Wahlstrom, I. Skog, and H. Peter, "Inertial Sensor Array Processing with Motion Models," in *21st International Conference on Information Fusion (FUSION)*, 2018, pp. 801–806.

[27] S. Sukkarieh, P. Gibbens, B. Grocholsky, K. Willis, and H. F. Durrant-Whyte, "A Low-Cost, Redundant Inertial Measurement Unit for Unmanned Air Vehicles," *The International Journal of Robotics Research*, vol. 19, no. 11, pp. 1089–1103, 2006.

[28] Y. N. Bezkorovainyi, "Improvement of UAV Positioning by Information of Inertial Sensors," in *2018 IEEE 5th International Conference on Methods and Systems of Navigation and Motion Control (MSNMC)*. IEEE, 2018, pp. 123–126.

[29] J. B. Bancroft, "Multiple Inertial Measurement Unit Integration for Pedestrian Navigation," Ph.D. dissertation, University of Calgary, 2010.

[30] R. G. Brown and P. Y. C. Hwang, *Introduction to random signals and applied Kalman filtering : with MATLAB exercises.* J. Wiley & Sons, 2012.

[31] N. Carlson, "Federated Filter for Fault-Tolerant Integrated Navigation Systems," in *IEEE PLANS '88.,Position Location and Navigation Symposium.* IEEE, 1988, pp. 110–119.

[32] J. B. Bancroft, G. Lachapelle, M. E. Cannon, and M. G. Petovello, "Twin IMU-HSGPS Integration for Pedestrian Navigation," in *ION GNSS 2008*, 2008, pp. 16–20.

[33] N. A. Carlson, "Federated Filter for Distributed Navigation and Tracking Applications," in *ION 58th Annual Meeting/CIGTF 21st Guidance Test Symposium, 24-26 June 2002, Albuquerque, NM*, 2002.

[34] M. Givens, C. Coopmans, and R. Christensen, "An Estimation Domain Approach to MEMS Multi-IMU Fusion for sUAS," in *2019 International Conference on Unmanned Aerial Systems (ICUAS).* Atlanta, GA: IEEE, 2019.

[35] KVH, "Guide to Comparing Gyro and IMU Technologies Micro-Electro-Mechanical Systems and Fiber Optic Gyros."

[36] K. Parsa, T. Lasky, and B. Ravani, "Design and mechatronic implementation of an accelerometer-based, kinematically redundant inertial measurement unit," in *Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.* IEEE, 2005, pp. 644–651.

[37] T. Moon, *Mathematical Methods and Algorithms for Signal Processing.* Pearson, 1999.

[38] P. Avery, "Combining Measurements with Correlated Errors," pp. 1–12, 1996.

# APPENDICES

# APPENDIX A

## Angular Resolution Derivation

Figure 2.1 is shown again below for convenience. A 2D pointing angle, representing the

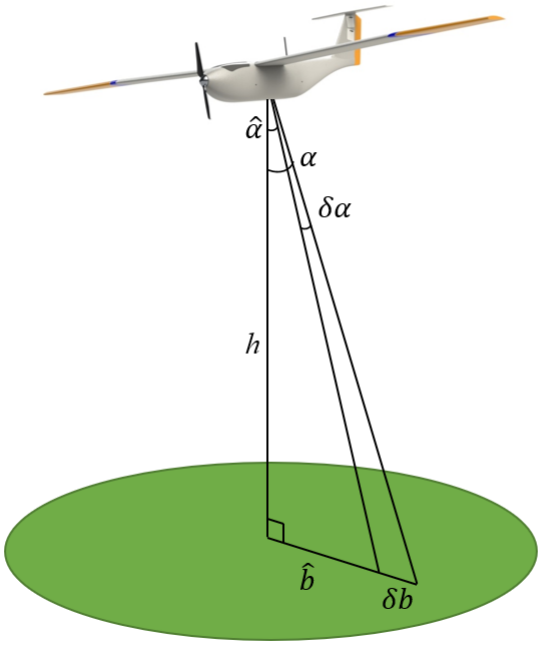


Fig. A.1: As $h$ increases, the effect that errors in the pointing angle $\delta\alpha$ have on the spatial resolution $\delta b$ increases linearly. See the appendix for derivation.

aircraft's angular distance from the vertical, $\alpha$, can be defined as

$$\alpha = \hat{\alpha} + \delta\alpha \tag{A.1}$$

where $\hat{\alpha}$ is the estimated angle and $\delta\alpha$ is the angular error. Likewise, the base of the triangle, $b$, can be defined as

$$b = \hat{b} + \delta b \tag{A.2}$$

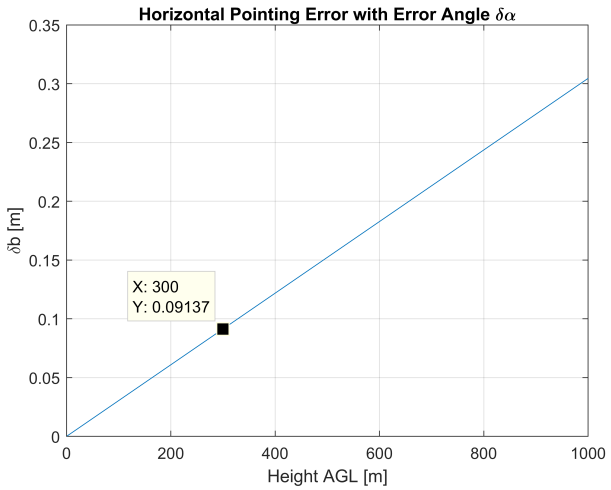


Fig. A.2: As $h$ increases, the spatial error $\delta b$ increases, reaching approximately 10cm at 300m altitude.

The height above ground can be related to the base of the triangle by

$$b = h\cos(\alpha) \tag{A.3}$$

$$\hat{b} = \hat{h}\cos(\hat{\alpha}) \tag{A.4}$$

Inserting equations A.3 and A.4 into A.2 results in

$$h\cos(\alpha) = \hat{h}\cos(\hat{\alpha}) + \delta b \tag{A.5}$$

Solving for $\delta b$ and substituting equation A.1 for $\hat{\alpha}$,

$$\boxed{\delta b = h\cos(\alpha) + h\cos(\alpha + \delta\alpha)} \tag{A.6}$$

For a given height AGL, a specified true value of $\alpha$, and a known angular resolution $\delta\alpha$, the spatial resolution, $\delta b$, can be obtained. Setting $\delta\alpha$ equal to 0.1 degrees and plotting the change in $\delta b$ as $h$ increases results in Fig. A.1.

# APPENDIX B

## Anomalous Covariance Results

In the centralized architecture, before and without calling on any sensor fusion algorithm, the variances of the estimated state errors for each IMU seem to decrease in magnitude. This change is not reflected in the standard deviations predicted by the covariance matrices, which are themselves unchanged from the independent case. While true of all three quantities, position, velocity, and attitude, the difference is most striking in the position solution:

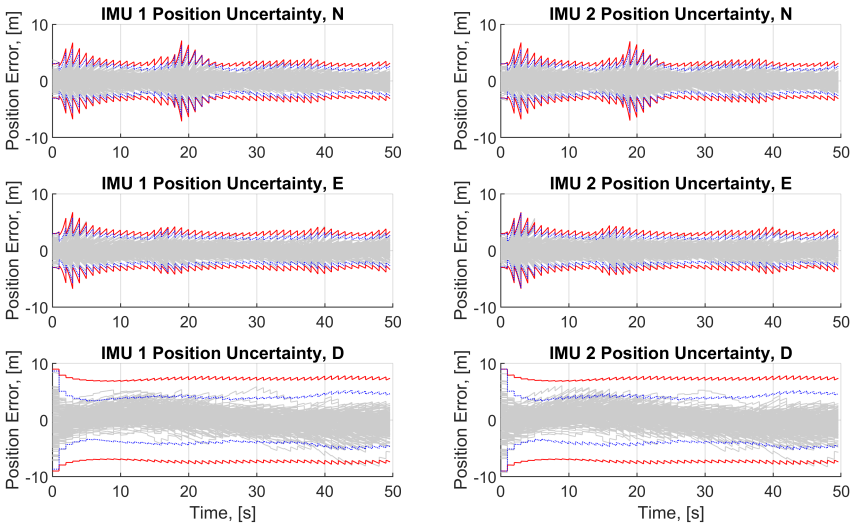


Fig. B.1: Ensemble position errors (blue dotted lines) do not match the predicted $3\sigma$ (solid red line).

The source(s) of these discrepancies is not known and is left to future investigation.

# APPENDIX C

## Error State Geometry Matrix Derivation

In Chapter 5, the derivation of the optimal constrained fusion algorithm required a matrix of geometry terms in order to move the solutions of each INS to a central point where they could be properly combined. For the total states, this could not be represented in matrix form because the transformation was nonlinear. However, it is possible to synthesize the $\mathbf{m}_i$ matrix for the error states. This appendix will show this short derivation for the position state and comment on its extension to the velocity and attitude states.

The definition of the true position of the IMU in relation to the true position state was given in Chapter 3:

$$\underline{r}_i^{ned} = \underline{r}_s^{ned} + \mathbf{R}_{b,i}^{ned} \underline{l}_i^b. \tag{C.1}$$

where, for this derivation the truth state $\underline{r}^{ned}$ is replaced with the location of the sensor fusion. In this thesis, these two are taken to be equal throughout but it is stated here for clarity. The same definition can be used for the estimated position:

$$\hat{\underline{r}}_i^{ned} = \hat{\underline{r}}_s^{ned} + \mathbf{R}_{b,i}^{ned} \underline{l}_i^b. \tag{C.2}$$

The position error state was also defined as

$$\underline{r}_i^{ned} = \hat{\underline{r}}_i^{ned} + \delta \underline{r}_i^{ned}. \tag{C.3}$$

The position error of the combined state is

$$\underline{r}_s^{ned} = \hat{\underline{r}}_s^{ned} + \delta \underline{r}_s^{ned}. \tag{C.4}$$

The error rotation matrix can be defined as a small rotation from the true rotation matrix,

$$\mathbf{R}_{b,i}^{ned} = \mathbf{R}_{\hat{b},i}^{ned}[\mathbf{I} - (\delta\underline{\theta}_i\times)]. \tag{C.5}$$

Inserting equations C.3-C.5 into equation C.1 yields

$$\hat{\underline{r}}_i^{ned} + \delta\underline{r}_i^{ned} = \hat{\underline{r}}_s^{ned} + \delta\underline{r}_s^{ned} + \mathbf{R}_{\hat{b},i}^{ned}[\mathbf{I} - (\delta\underline{\theta}_i\times)]\underline{l}_i^b. \tag{C.6}$$

The definition in C.2 can be substituted in and the equation simplified

$$\hat{\underline{r}}_s^{ned} + \mathbf{R}_{\hat{b},i}^{ned}\underline{l}_i^b + \delta\underline{r}_i^{ned} = \hat{\underline{r}}_s^{ned} + \delta\underline{r}_s^{ned} + \mathbf{R}_{\hat{b},i}^{ned}\underline{l}_i^b - \mathbf{R}_{\hat{b},i}^{ned}(\delta\underline{\theta}_i\times)\underline{l}_i^b \tag{C.7}$$

$$\delta\underline{r}_i^{ned} = \delta\underline{r}_s^{ned} - \mathbf{R}_{\hat{b},i}^{ned}(\delta\underline{\theta}\times)\underline{l}_i^b. \tag{C.8}$$

Using the identity $A \times B = -B \times A$ and solving for $\delta\underline{r}_s^{ned}$ yields the desired result:

$$\delta\underline{r}_s^{ned} = \delta\underline{r}_i^{ned} - \mathbf{R}_{\hat{b},i}^{ned}(\underline{l}_i^b\times)\delta\underline{\theta}_i. \tag{C.9}$$

The same process can be followed for velocity and attitude, yielding

$$\delta\underline{v}_s^{ned} = \delta\underline{v}_i^{ned} - \mathbf{R}_{\hat{b},i}^{ned}((\underline{\omega}_i \times \underline{l}_i^b)\times)\delta\underline{\theta}_i \tag{C.10}$$

$$\delta\underline{\theta}_s = \delta\underline{\theta}_i \tag{C.11}$$

which agree with equation 5.8.

# APPENDIX D

## Detailed Results of OCFA Geometry Study

Table D.1: Results of OCFA Geometry Study, uncorrelated measurements, 0 meter lever arms

| Description | 1-IMU | 2-IMU | 3-IMU | 4-IMU | 5-IMU | 6-IMU | 7-IMU | Units |
|---|---|---|---|---|---|---|---|---|
| Mean of ensemble $3\sigma$ position, N | 2.6522 | 1.8793 | 1.5881 | 1.3324 | 1.2147 | 1.1124 | 1.0146 | m |
| Mean of ensemble $3\sigma$ position, E | 2.5591 | 1.8135 | 1.7008 | 1.2831 | 1.2434 | 1.1250 | 1.0276 | m |
| Mean of ensemble $3\sigma$ position, D | 4.4557 | 3.4031 | 3.4211 | 2.5003 | 2.5451 | 2.3216 | 2.1116 | m |
| Mean of covariance matrix $3\sigma$ position, N | 2.6683 | 1.8855 | 1.5875 | 1.3331 | 1.2149 | 1.1053 | 1.0212 | m |
| Mean of covariance matrix $3\sigma$ position, E | 2.5637 | 1.8137 | 1.6773 | 1.2828 | 1.2415 | 1.1200 | 1.0274 | m |
| Mean of covariance matrix $3\sigma$ position, D | 4.2209 | 3.0631 | 3.3870 | 2.4290 | 2.4391 | 2.2928 | 2.0457 | m |
| Mean of ensemble $3\sigma$ velocity, N | 1.3743 | 0.9840 | 0.8768 | 0.6966 | 0.6558 | 0.5958 | 0.5395 | m/s |
| Mean of ensemble $3\sigma$ velocity, E | 1.1877 | 0.8451 | 0.8147 | 0.5969 | 0.5868 | 0.5309 | 0.4866 | m/s |
| Mean of ensemble $3\sigma$ velocity, D | 0.6550 | 0.7878 | 1.0696 | 0.6765 | 0.7962 | 0.6984 | 0.6626 | m/s |
| Mean of covariance matrix $3\sigma$ velocity, N | 1.3868 | 0.9797 | 0.8678 | 0.6923 | 0.6521 | 0.5897 | 0.5433 | m/s |
| Mean of covariance matrix $3\sigma$ velocity, E | 1.1916 | 0.8444 | 0.7945 | 0.5978 | 0.5864 | 0.5292 | 0.4844 | m/s |
| Mean of covariance matrix $3\sigma$ velocity, D | 0.5863 | 0.5738 | 1.0128 | 0.6664 | 0.7525 | 0.7556 | 0.6427 | m/s |
| Mean of ensemble $3\sigma$ attitude, $\theta_1$ | 0.0459 | 0.0333 | 0.0331 | 0.0232 | 0.0242 | 0.0213 | 0.0196 | rad |
| Mean of ensemble $3\sigma$ attitude, $\theta_2$ | 0.0462 | 0.0331 | 0.0357 | 0.0243 | 0.0258 | 0.0229 | 0.0207 | rad |
| Mean of ensemble $3\sigma$ attitude, $\theta_3$ | 0.1111 | 0.0838 | 0.0803 | 0.0597 | 0.0622 | 0.0515 | 0.0497 | rad |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_1$ | 0.0469 | 0.0331 | 0.0308 | 0.0234 | 0.0228 | 0.0205 | 0.0188 | rad |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_2$ | 0.0466 | 0.0330 | 0.0314 | 0.0233 | 0.0231 | 0.0208 | 0.0191 | rad |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_3$ | 0.1199 | 0.0887 | 0.0930 | 0.0698 | 0.0693 | 0.0654 | 0.0588 | rad |

Table D.2: Results of OCFA Geometry Study, correlated measurements, 0 meter lever arms

| Description | 1-IMU | 2-IMU | 3-IMU | 4-IMU | 5-IMU | 6-IMU | 7-IMU | Units |
|---|---|---|---|---|---|---|---|---|
| Mean of ensemble $3\sigma$ position, N | 2.6522 | 2.6364 | 2.7034 | 2.6382 | 2.5457 | 2.6958 | 2.5874 | m |
| Mean of ensemble $3\sigma$ position, E | 2.5591 | 2.6028 | 2.5286 | 2.5729 | 2.5313 | 2.6268 | 2.6116 | m |
| Mean of ensemble $3\sigma$ position, D | 4.4557 | 4.1596 | 4.2521 | 4.0724 | 3.7582 | 3.6807 | 3.6636 | m |
| Mean of covariance matrix $3\sigma$ position, N | 2.6683 | 2.0453 | 1.9087 | 1.7598 | 1.7259 | 1.6772 | 1.6412 | m |
| Mean of covariance matrix $3\sigma$ position, E | 2.5637 | 1.9744 | 2.0708 | 1.7108 | 1.7791 | 1.7102 | 1.6594 | m |
| Mean of covariance matrix $3\sigma$ position, D | 4.2209 | 3.3280 | 4.5864 | 3.3844 | 3.6654 | 3.5221 | 3.3004 | m |
| Mean of ensemble $3\sigma$ velocity, N | 1.3743 | 1.2294 | 1.2248 | 1.0757 | 1.0307 | 1.0406 | 0.9602 | m/s |
| Mean of ensemble $3\sigma$ velocity, E | 1.1877 | 1.1075 | 1.0556 | 0.9591 | 0.9567 | 0.9539 | 0.9091 | m/s |
| Mean of ensemble $3\sigma$ velocity, D | 0.6550 | 0.7574 | 0.9725 | 0.6619 | 0.7916 | 0.7299 | 0.7031 | m/s |
| Mean of covariance matrix $3\sigma$ velocity, N | 1.3868 | 1.0324 | 0.9717 | 0.8277 | 0.8130 | 0.7670 | 0.7319 | m/s |
| Mean of covariance matrix $3\sigma$ velocity, E | 1.1916 | 0.8901 | 0.9112 | 0.7229 | 0.7430 | 0.7007 | 0.6666 | m/s |
| Mean of covariance matrix $3\sigma$ velocity, D | 0.5863 | 0.4886 | 1.3692 | 0.9499 | 1.0260 | 1.0440 | 0.9222 | m/s |
| Mean of ensemble $3\sigma$ attitude, $\theta_1$ | 0.0459 | 0.0368 | 0.0356 | 0.0293 | 0.0286 | 0.0277 | 0.0260 | rad |
| Mean of ensemble $3\sigma$ attitude, $\theta_2$ | 0.0462 | 0.0393 | 0.0438 | 0.0312 | 0.0345 | 0.0332 | 0.0298 | rad |
| Mean of ensemble $3\sigma$ attitude, $\theta_3$ 0.1111 | 0.0930 | 0.0853 | 0.0673 | 0.0711 | 0.0687 | 0.0602 | rad | |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_1$ | 0.0469 | 0.0339 | 0.0322 | 0.0253 | 0.0249 | 0.0229 | 0.0213 | rad |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_2$ | 0.0466 | 0.0341 | 0.0336 | 0.0262 | 0.0268 | 0.0247 | 0.0232 | rad |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_3$ | 0.1199 | 0.0869 | 0.0985 | 0.0779 | 0.0763 | 0.0725 | 0.0657 | rad |

Table D.3: Results of OCFA Geometry Study, uncorrelated measurements, 1 meter lever arms

| Description | 1-IMU | 2-IMU | 3-IMU, triangle | 3-IMU, line | 4-IMU | 5-IMU | 6-IMU | 7-IMU | Units |
|---|---|---|---|---|---|---|---|---|---|
| Mean of ensemble $3\sigma$ position, N | 2.6533 | 1.8826 | 1.5946 | 1.5896 | 1.3351 | 1.2296 | 1.1148 | 1.0155 | m |
| Mean of ensemble $3\sigma$ position, E | 2.5597 | 1.8200 | 1.6818 | 1.6999 | 1.2864 | 1.2512 | 1.1266 | 1.0309 | m |
| Mean of ensemble $3\sigma$ position, D | 4.4542 | 3.5159 | 3.5088 | 3.4811 | 2.6477 | 2.7026 | 2.4619 | 2.1674 | m |
| Mean of covariance matrix $3\sigma$ position, N | 2.6774 | 1.8955 | 1.5939 | 1.5908 | 1.3380 | 1.2196 | 1.1074 | 1.0246 | m |
| Mean of covariance matrix $3\sigma$ position, E | 2.5627 | 1.8246 | 1.6817 | 1.6757 | 1.2936 | 1.2505 | 1.1265 | 1.0350 | m |
| Mean of covariance matrix $3\sigma$ position, D | 4.2202 | 3.1895 | 3.3600 | 3.4406 | 2.4666 | 2.7954 | 2.4214 | 2.0908 | m |
| Mean of ensemble $3\sigma$ velocity, N | 1.3752 | 0.9851 | 0.8775 | 0.8778 | 0.6981 | 0.6551 | 0.5963 | 0.5400 | m/s |
| Mean of ensemble $3\sigma$ velocity, E | 1.1891 | 0.8490 | 0.8053 | 0.8144 | 0.5989 | 0.5922 | 0.5318 | 0.4890 | m/s |
| Mean of ensemble $3\sigma$ velocity, D | 0.6552 | 0.9236 | 1.1152 | 1.0855 | 0.8293 | 0.8852 | 0.7923 | 0.7335 | m/s |
| Mean of covariance matrix $3\sigma$ velocity, N | 1.3883 | 0.9805 | 0.8637 | 0.8672 | 0.6928 | 0.6523 | 0.5903 | 0.5438 | m/s |
| Mean of covariance matrix $3\sigma$ velocity, E | 1.1927 | 0.8478 | 0.7967 | 0.7946 | 0.5996 | 0.5920 | 0.5290 | 0.4855 | m/s |
| Mean of covariance matrix $3\sigma$ velocity, D | 0.5866 | 0.6907 | 1.0078 | 1.0025 | 0.7120 | 1.0568 | 0.8480 | 0.6990 | m/s |
| Mean of ensemble $3\sigma$ attitude, $\theta_1$ | 0.0460 | 0.0335 | 0.0328 | 0.0329 | 0.0234 | 0.0247 | 0.0216 | 0.0198 | rad |
| Mean of ensemble $3\sigma$ attitude, $\theta_2$ | 0.0462 | 0.0336 | 0.0367 | 0.0365 | 0.0248 | 0.0258 | 0.0235 | 0.0207 | rad |
| Mean of ensemble $3\sigma$ attitude, $\theta_3$ | 0.1118 | 0.0846 | 0.0837 | 0.0799 | 0.0605 | 0.0638 | 0.0524 | 0.0503 | rad |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_1$ | 0.0469 | 0.0331 | 0.0308 | 0.0307 | 0.0234 | 0.0228 | 0.0205 | 0.0188 | rad |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_2$ | 0.0466 | 0.0330 | 0.0311 | 0.0312 | 0.0236 | 0.0233 | 0.0209 | 0.0192 | rad |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_3$ | 0.1195 | 0.0915 | 0.0902 | 0.0930 | 0.0697 | 0.0774 | 0.0677 | 0.0592 | rad |

Table D.4: Results of OCFA Geometry Study, correlated measurements, 1 meter lever arms

| Description | 1-IMU | 2-IMU | 3-IMU, triangle | 3-IMU, line | 4-IMU | 5-IMU | 6-IMU | 7-IMU | Units |
|---|---|---|---|---|---|---|---|---|---|
| Mean of ensemble $3\sigma$ position, N | 2.6533 | 2.6363 | 2.5727 | 2.6028 | 2.6428 | 2.5461 | 2.7006 | 2.5935 | m |
| Mean of ensemble $3\sigma$ position, E | 2.5597 | 2.6020 | 2.5651 | 2.6331 | 2.5767 | 2.5295 | 2.6274 | 2.6156 | m |
| Mean of ensemble $3\sigma$ position, D | 4.4542 | 4.1676 | 4.0266 | 4.1166 | 4.0859 | 3.7638 | 3.6906 | 3.6689 | m |
| Mean of covariance matrix $3\sigma$ position, N | 2.6774 | 2.0528 | 1.9229 | 1.9170 | 1.7633 | 1.7291 | 1.6784 | 1.6421 | m |
| Mean of covariance matrix $3\sigma$ position, E | 2.5627 | 1.9846 | 2.0662 | 2.0645 | 1.7191 | 1.7844 | 1.7132 | 1.6613 | m |
| Mean of covariance matrix $3\sigma$ position, D | 4.2202 | 3.5078 | 4.4262 | 4.6403 | 3.4932 | 3.7670 | 3.6661 | 3.3856 | m |
| Mean of ensemble $3\sigma$ velocity, N | 1.3752 | 1.2297 | 1.1723 | 1.1677 | 1.0776 | 1.0305 | 1.0419 | 0.9622 | m/s |
| Mean of ensemble $3\sigma$ velocity, E | 1.1891 | 1.1071 | 1.0669 | 1.0902 | 0.9603 | 0.9566 | 0.9522 | 0.9107 | m/s |
| Mean of ensemble $3\sigma$ velocity, D | 0.6552 | 0.8476 | 1.0123 | 0.9711 | 0.7455 | 0.8292 | 0.7960 | 0.7462 | m/s |
| Mean of covariance matrix $3\sigma$ velocity, N | 1.3883 | 1.0331 | 0.9802 | 0.9793 | 0.8281 | 0.8132 | 0.7674 | 0.7321 | m/s |
| Mean of covariance matrix $3\sigma$ velocity, E | 1.1927 | 0.8917 | 0.9028 | 0.9057 | 0.7230 | 0.7436 | 0.6995 | 0.6655 | m/s |
| Mean of covariance matrix $3\sigma$ velocity, D | 0.5866 | 0.6797 | 1.1980 | 1.3611 | 1.0496 | 1.1187 | 1.1401 | 0.9780 | m/s |
| Mean of ensemble $3\sigma$ attitude, $\theta_1$ | 0.0460 | 0.0369 | 0.0371 | 0.0381 | 0.0295 | 0.0286 | 0.0279 | 0.0262 | rad |
| Mean of ensemble $3\sigma$ attitude, $\theta_2$ | 0.0462 | 0.0396 | 0.0424 | 0.0427 | 0.0316 | 0.0347 | 0.0335 | 0.0303 | rad |
| Mean of ensemble $3\sigma$ attitude, $\theta_3$ | 0.1118 | 0.0932 | 0.0872 | 0.0885 | 0.0682 | 0.0710 | 0.0692 | 0.0611 | rad |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_1$ | 0.0469 | 0.0338 | 0.0321 | 0.0321 | 0.0253 | 0.0249 | 0.0229 | 0.0213 | rad |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_2$ | 0.0466 | 0.0341 | 0.0337 | 0.0339 | 0.0263 | 0.0269 | 0.0247 | 0.0232 | rad |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_3$ | 0.1195 | 0.0895 | 0.0921 | 0.1005 | 0.0784 | 0.0770 | 0.0750 | 0.0669 | rad |

Table D.5: Results of OCFA Geometry Study, uncorrelated measurements, 3 meter lever arms

| Description | 1-IMU | 2-IMU | 3-IMU, triangle | 3-IMU, line | 4-IMU | 5-IMU | 6-IMU | 7-IMU | Units |
|---|---|---|---|---|---|---|---|---|---|
| Mean of ensemble $3\sigma$ position, N | 2.6533 | 2.6363 | 2.5727 | 2.6028 | 2.6428 | 2.5461 | 2.7006 | 2.5935 | m |
| Mean of ensemble $3\sigma$ position, E | 2.5597 | 2.6020 | 2.5651 | 2.6331 | 2.5767 | 2.5295 | 2.6274 | 2.6156 | m |
| Mean of ensemble $3\sigma$ position, D | 4.4542 | 4.1676 | 4.0266 | 4.1166 | 4.0859 | 3.7638 | 3.6906 | 3.6689 | m |
| Mean of covariance matrix $3\sigma$ position, N | 2.6774 | 2.0528 | 1.9229 | 1.9170 | 1.7633 | 1.7291 | 1.6784 | 1.6421 | m |
| Mean of covariance matrix $3\sigma$ position, E | 2.5627 | 1.9846 | 2.0662 | 2.0645 | 1.7191 | 1.7844 | 1.7132 | 1.6613 | m |
| Mean of covariance matrix $3\sigma$ position, D | 4.2202 | 3.5078 | 4.4262 | 4.6403 | 3.4932 | 3.7670 | 3.6661 | 3.3856 | m |
| Mean of ensemble $3\sigma$ velocity, N | 1.3752 | 1.2297 | 1.1723 | 1.1677 | 1.0776 | 1.0305 | 1.0419 | 0.9622 | m/s |
| Mean of ensemble $3\sigma$ velocity, E | 1.1891 | 1.1071 | 1.0669 | 1.0902 | 0.9603 | 0.9566 | 0.9522 | 0.9107 | m/s |
| Mean of ensemble $3\sigma$ velocity, D | 0.6552 | 0.8476 | 1.0123 | 0.9711 | 0.7455 | 0.8292 | 0.7960 | 0.7462 | m/s |
| Mean of covariance matrix $3\sigma$ velocity, N | 1.3883 | 1.0331 | 0.9802 | 0.9793 | 0.8281 | 0.8132 | 0.7674 | 0.7321 | m/s |
| Mean of covariance matrix $3\sigma$ velocity, E | 1.1927 | 0.8917 | 0.9028 | 0.9057 | 0.7230 | 0.7436 | 0.6995 | 0.6655 | m/s |
| Mean of covariance matrix $3\sigma$ velocity, D | 0.5866 | 0.6797 | 1.1980 | 1.3611 | 1.0496 | 1.1187 | 1.1401 | 0.9780 | m/s |
| Mean of ensemble $3\sigma$ attitude, $\theta_1$ | 0.0460 | 0.0369 | 0.0371 | 0.0381 | 0.0295 | 0.0286 | 0.0279 | 0.0262 | rad |
| Mean of ensemble $3\sigma$ attitude, $\theta_2$ | 0.0462 | 0.0396 | 0.0424 | 0.0427 | 0.0316 | 0.0347 | 0.0335 | 0.0303 | rad |
| Mean of ensemble $3\sigma$ attitude, $\theta_3$ | 0.1118 | 0.0932 | 0.0872 | 0.0885 | 0.0682 | 0.0710 | 0.0692 | 0.0611 | rad |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_1$ | 0.0469 | 0.0338 | 0.0321 | 0.0321 | 0.0253 | 0.0249 | 0.0229 | 0.0213 | rad |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_2$ | 0.0466 | 0.0341 | 0.0337 | 0.0339 | 0.0263 | 0.0269 | 0.0247 | 0.0232 | rad |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_3$ | 0.1195 | 0.0895 | 0.0921 | 0.1005 | 0.0784 | 0.0770 | 0.0750 | 0.0669 | rad |

Table D.6: Results of OCFA Geometry Study, correlated measurements, 3 meter lever arms

| Description | 1-IMU | 2-IMU | 3-IMU, triangle | 3-IMU, line | 4-IMU | 5-IMU | 6-IMU | 7-IMU | Units |
|---|---|---|---|---|---|---|---|---|---|
| Mean of ensemble $3\sigma$ position, N | 2.6644 | 2.6448 | 2.5955 | 2.6124 | 2.6602 | 2.5545 | 2.7181 | 2.6130 | m |
| Mean of ensemble $3\sigma$ position, E | 2.5779 | 2.5351 | 2.5776 | 2.6450 | 2.5919 | 2.5325 | 2.6370 | 2.6301 | m |
| Mean of ensemble $3\sigma$ position, D | 4.5128 | 4.3123 | 3.9621 | 4.1546 | 4.1557 | 3.8145 | 3.7482 | 3.7052 | m |
| Mean of covariance matrix $3\sigma$ position, N | 2.7067 | 2.1162 | 1.9599 | 1.9436 | 1.7949 | 1.7538 | 1.6922 | 1.6542 | m |
| Mean of covariance matrix $3\sigma$ position, E | 2.5787 | 2.0595 | 2.1297 | 2.0893 | 1.7802 | 1.8280 | 1.7574 | 1.6935 | m |
| Mean of covariance matrix $3\sigma$ position, D | 4.2181 | 4.5503 | 5.4020 | 5.1071 | 4.4855 | 4.5084 | 4.3992 | 3.9577 | m |
| Mean of ensemble $3\sigma$ velocity, N | 1.3844 | 1.2441 | 1.1684 | 1.1752 | 1.0843 | 1.0327 | 1.0494 | 0.9706 | m/s |
| Mean of ensemble $3\sigma$ velocity, E | 1.2215 | 1.0742 | 1.0694 | 1.0947 | 0.9675 | 0.9606 | 0.9551 | 0.9196 | m/s |
| Mean of ensemble $3\sigma$ velocity, D | 0.6756 | 1.2013 | 1.2982 | 1.2030 | 1.1336 | 1.0965 | 1.0846 | 0.9726 | m/s |
| Mean of covariance matrix $3\sigma$ velocity, N | 1.4028 | 1.0303 | 0.9752 | 0.9795 | 0.8304 | 0.8147 | 0.7702 | 0.7339 | m/s |
| Mean of covariance matrix $3\sigma$ velocity, E | 1.1841 | 0.9061 | 0.9113 | 0.9094 | 0.7321 | 0.7522 | 0.7062 | 0.6707 | m/s |
| Mean of covariance matrix $3\sigma$ velocity, D | 0.5846 | 1.6165 | 1.9435 | 1.6497 | 1.6443 | 1.6088 | 1.5651 | 1.3157 | m/s |
| Mean of ensemble $3\sigma$ attitude, $\theta_1$ | 0.0482 | 0.0374 | 0.0373 | 0.0383 | 0.0304 | 0.0291 | 0.0286 | 0.0269 | rad |
| Mean of ensemble $3\sigma$ attitude, $\theta_2$ | 0.0463 | 0.0409 | 0.0439 | 0.0450 | 0.0336 | 0.0358 | 0.0352 | 0.0321 | rad |
| Mean of ensemble $3\sigma$ attitude, $\theta_3$ | 0.1233 | 0.0985 | 0.0881 | 0.0897 | 0.0729 | 0.0728 | 0.0718 | 0.0644 | rad |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_1$ | 0.0465 | 0.0336 | 0.0320 | 0.0320 | 0.0252 | 0.0248 | 0.0228 | 0.0213 | rad |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_2$ | 0.0468 | 0.0345 | 0.0339 | 0.0338 | 0.0270 | 0.0274 | 0.0255 | 0.0239 | rad |
| Mean of covariance matrix $3\sigma$ attitude, $\theta_3$ | 0.1176 | 0.1077 | 0.1046 | 0.1064 | 0.0855 | 0.0834 | 0.0838 | 0.0734 | rad |