AUTONOMOUS AND RESILIENT MANAGEMENT OF ALL-SOURCE SENSORS FOR NAVIGATION INTEGRITY: A COMPARISON AND ANALYSIS

THESIS

Niles A. Tate, First Lieutenant, USAF

AFIT-ENG-MS-22-M-065

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-MS-22-M-065

AUTONOMOUS AND RESILIENT MANAGEMENT OF ALL-SOURCE SENSORS FOR NAVIGATION INTEGRITY: A COMPARISON AND ANALYSIS

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Niles A. Tate, B.S.E.C.E.

First Lieutenant, USAF

March 24, 2022

AFIT-ENG-MS-22-M-065

AUTONOMOUS AND RESILIENT MANAGEMENT OF ALL-SOURCE SENSORS FOR NAVIGATION INTEGRITY: A COMPARISON AND ANALYSIS

THESIS

Niles A. Tate, B.S.E.C.E.
First Lieutenant, USAF

Committee Membership:

Robert C. Leishman, Ph.D
Chair

Meir N. Pachter, Ph.D
Member

Clark N. Taylor, Ph.D
Member

Abstract

When navigating using Global Navigation Satellite Systems (GNSS), multiple/redundant, synchronous pseudorange measurements are readily available. This provides an accurate navigation solution. However, when navigating in a GNSS degraded and/or denied region, this is not guaranteed. This is tantamount to a low level of integrity when using a total system to provide a navigation solution. In response to this challenge, the Air Force Institute of Technology (AFIT) Autonomy and Navigation Technology (ANT) Center developed a framework known as Autonomous and Resilient Management of All-source Sensors (ARMAS). The ARMAS framework is designed to be resilient towards data corruption caused from mismodeled, uncalibrated, and faulty sensors, by employing: sensor validation, a Fault Detection and Exclusion (FDE) function, recalibration, and remodeling modes in a single integrated architecture. This thesis further expands on the ANT Center ARMAS work by performing a comparison against an incumbent Residual-Based Receiver Autonomous Integrity Monitoring (RBRAIM) scheme, a widely popular integrity monitoring method. Using both simulated and flight data, different GNSS "fault" scenarios are presented to both systems. From there, the ARMAS and RBRAIM systems' performance is evaluated with respect to accuracy, reliability, and effectiveness. During jamming scenarios, ARMAS outperformed RBRAIM in accuracy by a Root Mean Square Error (RMSE) value of 0.11 when predicting the 3D position of the receiver. For the biasing scenarios, RBRAIM thoroughly outperformed ARMAS in accuracy by a difference of 34.39 in RMSE. Additionally, RBRAIM reliably and effectively handles the bias compared to ARMAS. This thesis provides quantitative data showing that ARMAS performs better than RBRAIM for jamming navigation scenarios while RBRAIM performs

better than ARMAS in biasing scenarios.

Acknowledgements

I am very grateful to AFIT and the USAF for allowing me to participate in this challenging but rewarding opportunity.

Thank you to the professors who dedicated their time and effort. I would like to thank my family, friends, and previous coworkers for supporting and cheering me on. I would not have survived if it wasn't for a number of classmates and friends I met during this program. I am truly thankful to you all.

I would like to especially thank my parents for believing in me when I didn't believe in myself!

Niles A. Tate

Table of Contents

Page

List of Figures

Figure Page

List of Tables

AUTONOMOUS AND RESILIENT MANAGEMENT OF ALL-SOURCE SENSORS FOR NAVIGATION INTEGRITY: A COMPARISON AND ANALYSIS

I. Introduction

1.1 Overview/Background

For more than 30 years, the satellite-based GPS has been used to provide Position, Navigation, and Timing (PNT) data to Department of Defense (DoD) users worldwide. According to the United States Government Accountability Office (GAO) Technological Assessment on Defense Navigation Capabilities presented May 2021, the DoD faces many challenges in developing and integrating alternative PNT technologies by not making it a priority. In the report, they expressed the need for new PNT technologies by stating "DoD's continued reliance on GPS, despite known GPS vulnerabilities to disruption, present a challenge for obtaining sufficient support to develop viable alternatives" [1]. Considering this, the ANT Center has developed a framework known as ARMAS to ensure the integrity and provide resilient navigation using Global Position System (GPS). The thesis expands on the work done on ARMAS by comparing and analyzing its performance to an incumbent Receiver Autonomous Integrity Monitoring (RAIM) system. This system is based off the work completed by [2] and [3]. Testing these systems in different "disrupted" scenarios, allows for a quantitative evaluation and demonstration of their performance with respect to accuracy, reliability, and functionality. The information provided by these tests advocates for the further pursuit to increase the reliability of GPS and viable alternatives in the PNT field.

1.2 Problem Statement

Build proven integrity monitoring methods of the navigation function to demonstrate and evaluate the capabilities of other systems handling challenging navigation scenarios compared to ARMAS.

1.3 Research Goals

- Design a high performance and reliable RAIM system.
- Perform a controlled experiment: validate the performance of the RAIM system developed in the thesis by using simulated data.
- Examine the reliability and effectiveness of identifying the biased satellite/space vehicle (SV) in the RAIM system and the ARMAS framework.
- Measure the accuracy of the receiver's provided position from both systems during realistic navigation scenarios.
- Evaluate and compare the results provided by the two systems.

1.4 Approach

The experiments to determine the accuracy, reliability, and effectiveness of the ARMAS and RAIM system will be performed using both Matlab and Python. The RAIM system is built in Python and then validated by using simulated flight data of a Class 2 Unmanned Aerial Vehicle (UAV) to provide an accurate position of the receiver on board. After validation, real flight data from an UAV flown at Camp Atterbury, Indiana is used to tune and establish a baseline for both systems. The flight data is then corrupted to replicate two jamming and two biasing scenarios for the systems to autonomously manage. These scenarios are run in Python for the RAIM

system and in Matlab for ARMAS. The data provided from the jamming scenarios demonstrates the accuracy drop-off of both systems when denied multiple satellites. The data provided from the biasing scenarios show how resilient and effective both systems are when the navigation data is being corrupted.

1.5 Assumptions/Limitations

The RAIM and ARMAS system presented in this research have not been tested on board an aircraft, so the position state produced by the simulated systems may differ from the results produced by the real-world system. The clock bias used in the Camp Atterbury experiment is not the real clock bias but was generated from the difference of the predicted pseduorange measurement from the system compared to the actual pseudorange measurement from each satellite and then averaged. The RAIM system synthesized in this research does not have sensors implemented into the system like the ARMAS system has, leading to the ARMAS system receiving more information for the navigation state calculation. The ARMAS system is built on the assumption of a single fault/satellite failure causing degraded performance during multiple bias situations.

1.6 Research Contributions

- A proven, resilient RAIM system and its methods simulated in Python
- Demonstration and quantitative evaluation of each systems' performance during disruptive navigation scenarios using real flight data
- Comparison of the systems' performance, pointing to improvements and weak points of the ARMAS system

1.7 Thesis Overview

The thesis is organized as follows. Chapter II provides some background knowledge to better understand how ARMAS and RAIM obtain the navigation solution. Chapter III provides the methodology for building the RAIM system and the experimentation process for the evaluation of the performance of the RAIM and ARMAS systems. The results of the experimentation process outlined in Chapter III are provided in Chapter IV along with an analysis. The final chapter presents the conclusions and future work associated with this research, to further enhance the ARMAS system and the field of PNT.

II. Background

2.1 Overview

This chapter offers relevant knowledge and concepts to better understand RAIM, ARMAS navigation framework, and how they relate to obtaining the navigation solution. The chapter starts with an exposition of GPS point positioning, the discussion of reference frames, the GPS navigation message, the pseudorange concept, and Least Square Estimation (LSE) – all relevant to the thesis topic. Then, Inertial Navigation System (INS), the integration between INS and GPS navigation, and recursive model estimation is discussed to build a foundation for understanding the functionality of RAIM. Lastly, a brief synopsis of related research will be discussed.

2.2 Global Position System (GPS) Point Positioning

GPS is positioning based on multilateration, a method of determining position by measuring distances to points of known coordinates. Multilateration requires a minimum of 3 known points to calculate latitude, longitude, and altitude. However, to use GPS point position, a minimum of 4 measurements from 4 satellites need to be available for the estimation of the GPS receiver's clock error. To receive these measurements, the position of the satellites must be known – the ephemeris. The following subsections will provide tools to better understand how GPS point positioning is obtained.

2.2.1 Reference Frames

A vehicle's position, velocity, and attitude must be expressed with respect to a reference frame. A more detailed explanation of the navigation reference frames, and

their purposes is provided in [4]. This subsection will discuss the navigation reference frames necessary for this research.

Earth-centered Inertial frame (i-frame) – is a non-rotating reference frame with the center of the Earth as its origin. The x and y plane are located on the Equator's plane with +x axis pointing towards the first star in the constellation of Aries. The z-axis is aligned with the Earth's poles with +z axis pointing northward to the North Pole.

Earth-centered Earth-fixed frame (e-frame) – is a rotating reference frame attached to the Earth. The x and y plane are collocated with the Equatorial plane with +x pointing towards the Prime/Greenwich Meridian and +y axis pointing towards 90° East longitude. The z-axis runs in between the North and South Poles, with +z pointing northward.

Earth-fixed navigation frame (n-frame) – is a reference frame with its origin typically located near the surface of the Earth, known as North-East-Down (NED) or Local Tangent Plane (LTP). The +x direction point North, the +y direction points East, and the +z direction point Down with respect to the origin. The reference frame is typically used for tracking applications due to it being more intuitive and practical. There is a variant of NED called East-North-Up (ENU) that swaps the x and y-axis and negates the z-axis for orientation.

Aircraft body frame (b-frame) – is a reference frame with the origin fixed at the aircraft's center. The +x points out the nose, the +y points out the right wing, and the +z points out the bottom of the aircraft.

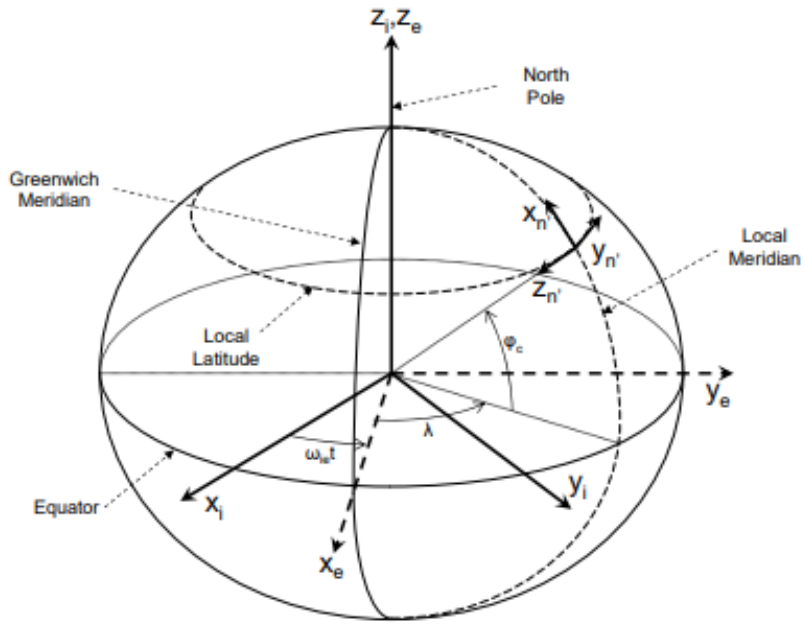


Figure 1: Earth-Centered Inertial, Earth-Centered Earth-Fixed, and Earth-Fixed Reference Frames [5]

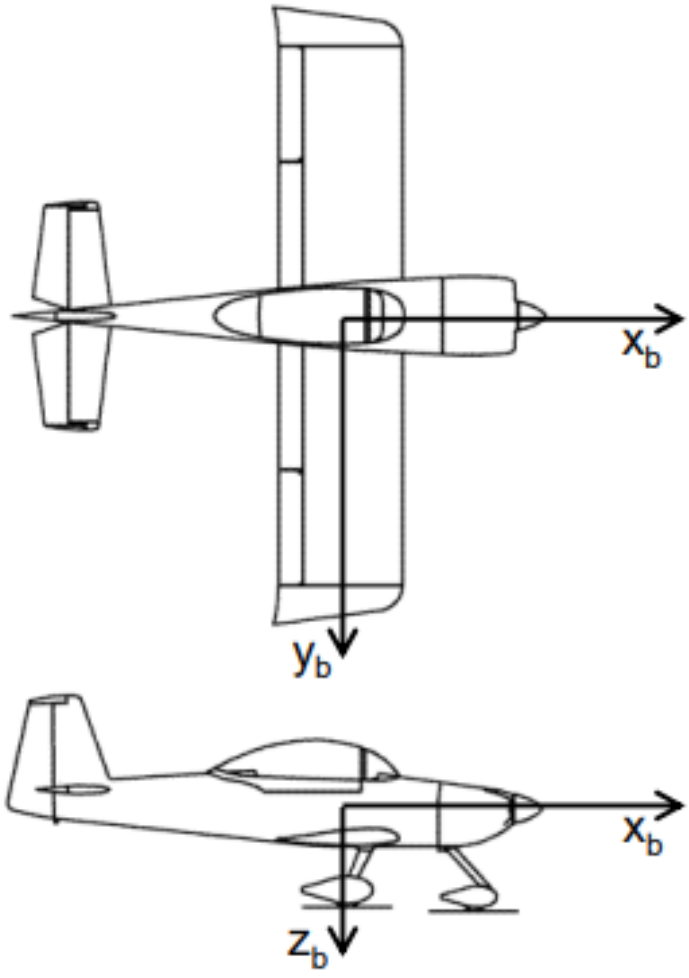


Figure 2: Aircraft Body Reference Frame [5]

2.2.2 Navigation Message

A signal from each satellite is encoded and transmitted in the "Navigation Message," which is read by the user's GPS receivers. The Navigation Message is data that provides the user with information necessary to compute the location/ephemeris of each visible satellite and the time of transmission for each navigation signal [6]. This subsection will outline the main features of the GPS navigation message. For a more complete description, refer to [7].

The GPS navigation message is transmitted in five 300-bit subframes with each subframe containing ten 30-bit words as shown in Figure 3. In each subframe, the telemetry (TLM) data is the first word shown. TLM provides a fixed 8-bit pattern that assists the user equipment in locating the beginning of each subframe. The second word shown in each subframe is a handover word (HOW). HOW allows the user equipment to convert C/A code tracking to P(Y) code tracking that provides the GPS time-of-week (TOW) and the subframe number.

Subframe 1 – yields the GPS transmission number that helps keep track of rollovers, satellite clock correction terms for precise ranging due to of lack of perfect synchronization between SV broadcast signals and GPS system time, issue of data clock (IODC) to point out the current set of navigation data, and SV health indicator.

Subframe 2&3 – provides the Keplerian orbital elements that allows the user equipment to determine satellite location and an issue of data ephemeris (IODE) that is used to detect changes in broadcast orbital elements.

Subframe 4 – contains almanac data that allow the determination of satellite positions to help with acquisition for SVs 1-32, navigation message correction table (NMCT) for range corrections, ionospheric correction parameters to relate Universal Time Coordinated (UTC) to GPS system time, and configuration and health flags for SVs 1-32.

Subframe 5 – includes almanac data like Subframe 4, but with time and week number included.

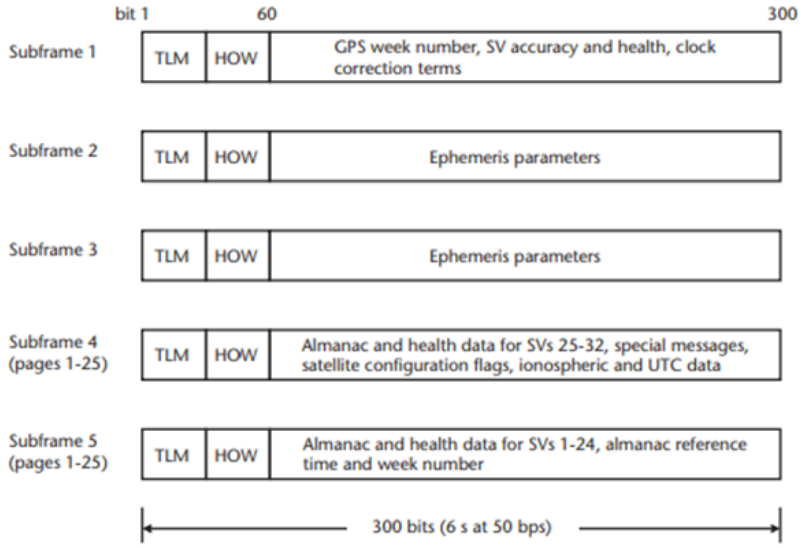


Figure 3: Navigation Message Format [6]

2.2.3 Pseudorange

When a satellite transmits the navigation message, the receiver attempts to measure the distance to the satellite. The time of the transmission is based on the atomic clock in the satellite and is encoded into the transmitted message. The receiver then records the time of when the message is received using its own clock and calculates its "distance" from the SV according to the difference in time (Figure 4). This is referred to as a pseudorange. Pseudorange is like range, but instead of just path delays there is also the effects of clock error due to the receiver clock not being 100% accurate. The equation can be written as:

$$P^{s} = (T - T^{s})c \tag{1}$$

where T is the known reading of the receiver clock when the signal is received, T^{s} is the reading of the satellite clock when the signal was transmitted, and c is the speed of light in a vacuum (299792458 m/s) [8].

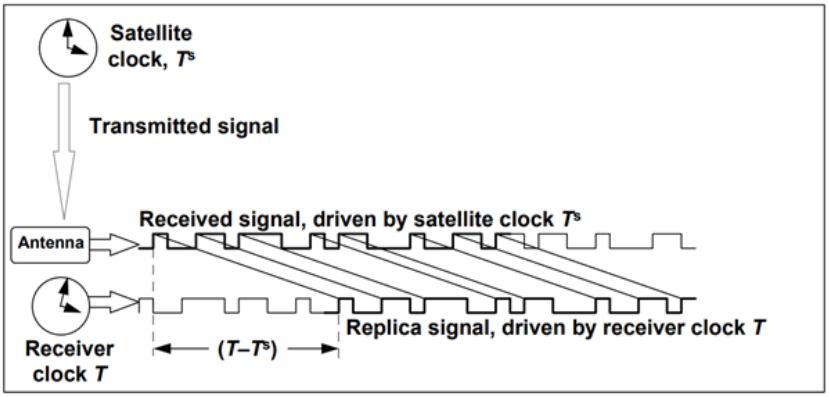


Figure 4: GPS Pseudorange in Relation to Satellite and Receiver Clocks [8]

From the figure above, the equation can further be developed by setting T equal to the true receive time plus a clock bias τ for both the receiver and satellite clocks:

$$T = t + \tau \tag{2}$$

$$T^s = t^s + \tau^s \tag{3}$$

When substituted into the original pseudorange equation:

$$\begin{aligned} P^s(t) &= ((t+\tau) - (t^s + \tau^s))\, c \\ &= (t - t^s)\, c + c\tau - c\tau^s \\ &= \rho^s(t, t^s) + c\tau - c\tau^s \end{aligned} \tag{4}$$

A correction due to the general theory of relativity is also implemented. From here,

the Pythagorean Theorem is used to solve for the range:

$$\rho^s\left(t, t^s\right) = \sqrt{\left(x^s\left(t^s\right) - x\left(t\right)\right)^2 + \left(y\left(t^s\right) - y\left(t\right)\right)^2 + \left(z^s\left(t^s\right) - z\left(t\right)\right)^2} \tag{5}$$

With the ranging equation substituted into the pseudorange function, the completed pseudorange equation is derived:

$$P^n = \sqrt{\left(x^n - x\right)^2 + \left(y^n - y\right)^2 + \left(z^n - z\right)^2} + c\tau - c\tau^n \tag{6}$$

with superscripts n meaning the specific SV. From the navigation message, the satellite position (x^n, y^n, z^n) and satellite clock bias τ^n is known. This leaves receiver position (x, y, z) and receiver clock error τ as unknowns. These four unknowns can be solved using Iterative Least Squares (ILS) methods. Five iteration steps will do.

2.2.4 Least Squares Estimation (LSE)

To resolve the point position issue, the pseudorange observation equations must be linearized before using least square methods. For a more detailed explanation of this approach, see reference [9].

Initial coordinates of the receiver (x_0, y_0, z_0) are typically known – set at the center of the Earth, so the receiver coordinate can be expressed as:

$$x_i = x_0 + \Delta x_i,\ y_i = y_0 + \Delta y_i,\ z_i = z_0 + \Delta z_i \tag{7}$$

with $(\Delta x_i, \Delta y_i, \Delta z_i)$ represented as the new unknowns. From there, (x_i, y_i, z_i) can be rewritten into:

$$f(x_i, y_i, z_i) = f(x_0 + \Delta x_i,\ y_0 + \Delta y_i,\ z_0 + \Delta z_i) \tag{8}$$

Applying Taylor series, the above equation can be expanded into:

$$f(x_i, y_i, z_i) = f\left(x_0, y_0, z_0\right) + \frac{\partial f\left(x_0, y_0, z_0\right)}{\partial x_0}\Delta x_i + \frac{\partial f\left(x_0, y_0, z_0\right)}{\partial y_0}\Delta y_i + \frac{\partial f\left(x_0, y_0, z_0\right)}{\partial z_0}\Delta z_i + \frac{1}{2!}\frac{\partial^2 f}{\partial x^2}... \tag{9}$$

Due to linear expressions, Taylor's expansion can be truncated, and the linearized expression is introduced:

$$\frac{\partial f(x_0, y_0, z_0)}{\partial x_0} = \frac{x_n - x_0}{\rho_0^n},\ \frac{\partial f(x_0, y_0, z_0)}{\partial y} = \frac{y_n - y_0}{\rho_0^n},\ \frac{\partial f(x_0, y_0, z_0)}{\partial y} = \frac{z_n - z}{\rho_0^n} \tag{10}$$

with n being the specific SV in view. This leads into the pseudorange equation with errors not included being written as:

$$P^n = \rho_0^n - \frac{x_n - x_0}{\rho_0^n}\Delta x_i - \frac{y_n - y_0}{\rho_0^n}\Delta y_i - \frac{z_n - z_0}{\rho_0^n}\Delta z_i + \delta t_i * c \tag{11}$$

and then simplified to:

$$l^j = a_{x_i}^n \Delta x_i + a_{y_i}^n \Delta y_i + a_{z_i}^n \Delta z_i + c\Delta t_i \tag{12}$$

$$l^j = P^n - \rho_0^n \tag{13}$$

$$a_{x_i}^n = -\frac{x_n - x_0}{\rho_0^n},\ a_{y_i}^n = -\frac{y_n - y_0}{\rho_0^n},\ a_{z_i}^n = -\frac{z_n - z_0}{\rho_0^n} \tag{14}$$

This allows the above formula to be expressed as a Linear Regression (LR):

$$\vec{L} = A\vec{X} \tag{15}$$

$$\begin{bmatrix} l^1 \\ l^2 \\ l^3 \\ \vdots \\ l^n \end{bmatrix} = \begin{bmatrix} a^1_{x_i} & a^1_{y_i} & a^1_{z_i} & c \\ a^2_{x_i} & a^2_{y_i} & a^2_{z_i} & c \\ a^3_{x_i} & a^3_{y_i} & a^3_{z_i} & c \\ \vdots & \vdots & \vdots & \vdots \\ a^n_{x_i} & a^n_{y_i} & a^n_{z_i} & c \end{bmatrix} \begin{bmatrix} \Delta x_i \\ \Delta y_i \\ \Delta z_i \\ \Delta t_i \end{bmatrix} \tag{16}$$

where $\vec{L}$ is a vector of n observations/measurements, A is a matrix (Regressor), and $\vec{X}$ is the vector of unknowns/parameter. Lastly, the solution of the LR can be obtained by the LSE method:

$$\vec{X} = (A^T A)^{-1} \vec{L} \tag{17}$$

which is how $(\Delta x_i, \Delta y_i, \Delta z_i, \Delta t_i)$ is found, and then iteratively used to find the receiver coordinates:

$$x_i = x_0 + \Delta x_i,\ y_i = y_0 + \Delta y_i,\ z_i = z_0 + \Delta z_i \tag{18}$$

2.3 Inertial Navigation

As mentioned above GPS is a reference-based system whereas an INS is a form of dead reckoning that uses accelerometer and gyroscopes along with a processor [10]. With the gyroscopes providing attitude information in 3 dimensions and the accelerometers providing velocity and position through integration, the processor can output the current position, velocity and attitude of the host vehicle based on the host's initial state (Figure 5).

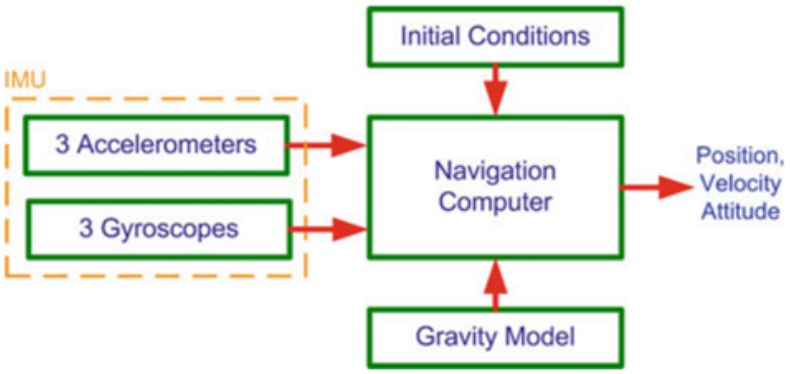


Figure 5: Block diagram of an INS [10]

2.4 Integrated INS/GPS Navigation

Due to the very different principles of operation of INS and GPS, they end up complementing each other (Table 1). Therefore, when data from both are combined, they can attain better performance than either of the systems can alone. Based on their complementary properties, INS and GPS are combined using an estimation technique, traditionally based on the Kalman filter, to obtain [11] (Figure 6):

1. Higher position and velocity accuracy.
2. Precise attitude information.
3. Higher data output rate.
4. Navigational solution during GPS signal blockages.

Characteristics	INS	GPS
Accuracy of navigational solution	Good short term accuracy which deteriorates with time	Good long term accuracy but noisy in short term
Initial conditions	Required	Not required
Attitude information	Available	Typically not available[a]
Sensitive to gravity	Yes	No
Self-contained	Yes	No
Jamming immunity	Yes	No
Output data rate	High	Low

[a] With multiple antennae, some GPS receivers can render attitude information as well

Table 1: Summary of Characteristics of INS and GPS [10]

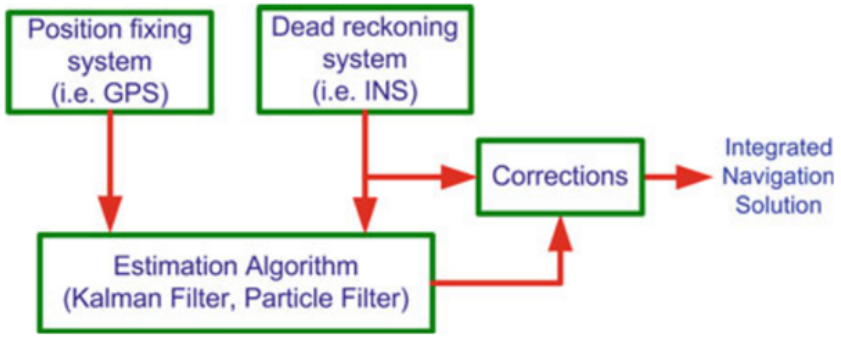


Figure 6: Diagram of Typical Integrated Navigation System [10]

2.5 Recursive Model Estimation

In modern navigation, the method of recursive model estimation is used to sequentially estimate the underlying state of a system (x_k) given the initial estimate (x_0) and measurements (z_k) obtained over time (k) [12]. The Bayesian filtering methods, such as the Kalman Filter (KF) algorithm, constructs the Probability Density Function (PDF) of the state based on all the available information, including received measurements [13]. Therefore, it forms the complete solution to the estimation problem. For linear Gaussian estimation problems, a KF is more suitable. However, for nonlinear

Gaussian estimation problems, an Extended Kalman Filter (EKF) is better suited for the task. For both filters, the algorithm consists of two main steps of prediction and update. For a more thorough explanation, please refer to [14].

The first stage of the KF algorithm is the prediction of the prior state estimate and prior error covariance. The predicted state estimate equation is written as:

$$\hat{x}_{\bar{k}} = F\hat{x}_{k-1} + Bu_{k-1} + w_{k-1} \tag{19}$$

To obtain the predicted state estimate ($\hat{x}_{\bar{k}}$), the state transition matrix (F) must be applied to the previously updated state estimate ($\hat{x}_{k-1}$). The state transition matrix is used to propagate the previous state estimate to a later time for the estimation of the state [15]. In addition, the control-input matrix (B) is applied to the control vector (u_{k-1}) and then followed by the process noise vector (w_{k-1}). The control-input matrix applied to the control vector determines the contribution the controls have on the change in state [16]. The process noise vector is typically assumed to be a zero-mean Gaussian with covariance (Q) which is the process uncertainty. The next prediction step is finding the predicted state error covariance:

$$P_{\bar{k}} = FP_{k-1}F^T + Q \tag{20}$$

The state error covariance in summary is the expectation of the variance between the predicted state and the true state. The error covariance has an important impact when it comes to the process of finding the state estimate.

The second stage of the KF algorithm is the update process once the new measurement has been received. It starts with the measurement residual being computed:

$$r_k = z_k - H\hat{x}_{\bar{k}} \tag{21}$$

The measurement residual or innovation is the difference between the true measurement (z_k) and the measurement matrix (H) applied to predicted state estimate. The purpose of the measurement matrix is to convert the state into a measurement. Therefore, when the H matrix is applied to the predicted state, it results in a predicted measurement. In conclusion, the measurement residual is the difference between the received measurement and the predicted measurement. From there, the Kalman gain is found:

$$S = HP_{\bar{k}}H^T + R \tag{22}$$

$$K_k = P_{\bar{k}}H^T(S)^{-1} \tag{23}$$

The Kalman gain weighs the predicted measurement and the received measurement, that is, determines how much trust should be put into either when forming the state estimate. The $P_{\bar{k}}H^T$ part of the equation represents the uncertainty of the prediction, while the residual's covariance (S) represents the uncertainty in the actual measurement. The measurement noise (R) is a zero-mean Gaussian with covariance applied to the residual's covariance equation. If the Kalman gain is high, then a higher weight is put on the actual measurement over the predicted. After obtaining the Kalman gain, the state estimate and error covariance can be updated:

$$\hat{x}_k = \hat{x}_{\bar{k}} + K_k r_k \tag{24}$$

$$P_k = (I - K_k H)P_{\bar{k}} \tag{25}$$

The updated state estimate ($\hat{x}_k$) is found by combining the predicted state estimate and the residual scaled by the Kalman gain. Applying the Kalman gain to the return difference/residual in addition to the prior state estimate, yields that updated state estimate. The updated error covariance, in summary, will either grow or shrink

in value from its pre-update status quantifying the uncertainty between truth and KF provided state estimate. The above equations are specific to the KF meaning some of these equations cannot be applied to an EKF. The predicted state estimate, predicted error covariance, measurement residual, and Kalman gain formulas are extended to handle a nonlinear dynamic system. To briefly state, the EKF takes nonlinear functions and linearizes them before completing the prediction of the state estimate and computation of the Kalman Gain.

2.6 Receiver Autonomous Integrity Monitoring (RAIM)

RAIM is a method used to provide a measure of trust which can be placed in the correctness of information supplied by the navigation system [17, 18, 19]. The RAIM technique monitors the integrity of navigation signals received independent of any external sources at any given time. RAIM algorithms use measurement redundancy to check consistency among the GPS measurements, by means of pseudorange residuals provided by the EKF algorithm and detects if the consistency check passes or fails [20]. If the check fails, the receiver provides an alert notifying the user that the integrity of the position is an issue. This method is known as Fault Detection (FD). For the FD function to run properly, a minimum of 5 satellites with low Geometric Dilution of Precision (GDOP) must be visible so that consistency checks can be performed. An upgraded method to FD is Fault Detection and Exclusion (FDE). The FDE function detects and identifies a faulty satellite to then exclude it from the navigation solution. Six satellites are required for the FDE function to be mechanized. After the FDE step, the position and Protection Level (PL) is calculated [21]. The PL calculation allows a statistical bound to be put on the positional error so that there is a measurement of risk that the alert limit has been surpassed. The PL is split between Horizontal Protection Level (HPL), which provides a bound on the horizontal position error with

a probability derived from the integrity requirements, and Vertical Protection Level (VPL), which does the same as HPL but for vertical position. RAIM is typically broken into two main groups: Measurement Rejection Approach (MRA) and Error Characterization Approach (ECA) [22, 23, 24, 25]. In the subsection below these methods will be briefly discussed. For more details on the subjects, please refer to [22, 23, 24].

2.6.1 Measurement Rejection Approach (MRA)

Typically, GNSS navigation assumes that integrity relies on the ability of the system to detect and exclude degraded measurements. This is the main basis for the measurement rejection approach. MRA is similar to the FDE concept. However, instead of the failure mode being specifically linked to a satellite issue or GDOP, it is understood as a large measurement error above a specified threshold no matter what the nature of the error is [22]. Therefore, this method utilizes solely fault-free measurements to calculate protection level. Meaning, that the fault-free measurements are used to compute the navigation solution guaranteed by the FDE function. This approach is used mainly for aviation purposes and not as much in urban environments due to many measurements being rejected from Non-Line of Sight (NLOS) multipath [25].

2.6.2 Error Characterization Approach (ECA)

The Error Characterization Approach does the opposite of MRA by characterizing measurement errors and then computing a protection level that protects against them. This approach allows a level of integrity in the system without the need to identify and remove degraded measurements, even if they are contaminated with very large errors [22]. Essentially, this approach does not require a FDE mechanism, as it calculates

the protection levels based on classification of faults in the satellite measurements. This method results in higher protection level in urban areas because it reflects the faults of NLOS multipath satellite signals in its navigation solution computation [26, 27, 28, 29].

2.7 Related Research

2.7.1 Autonomous and Resilient Management of All-source Sensors (ARMAS)

Introduced in 2018 as a method of addressing Navigation Warfare (NAVWAR), the Autonomy and Navigation Technology (ANT) Center provided a framework known as ARMAS. ARMAS is a framework used to manage heterogeneous, asynchronous all-source sensors [30] [31]. This subsection will provide a brief overview of the ARMAS framework, and is based on a more detailed explanation to be found in [30].

The framework is designed to be resilient towards any data corruption caused from mismodeled, uncalibrated, and faulty sensors, by combining, sensor validation, a FDE function, recalibration, and remodeling modes in a single architecture. This allows the framework to provide a higher probability of integrity when receiving navigation solutions. Using SCORPION, a pluggable Bayesian estimator that handles multiple concurrent state estimation subfilters, ARMAS employs pluggable EKF estimators to handle nonlinear navigation calculations [32]. The ARMAS framework has six modes that sensors could be categorized in: offline/online, monitor, validate, calibrate, remodel, or failed, as referenced in Figure 7.

Offline/Online Mode – after receiving the estimates, the sensors are initialized as either trusted or untrusted. Trusted sensors are directly brought online and pushed into monitoring mode. However, untrusted sensors are required to enter a sensor validation mode before pushed into monitoring mode.

Monitor Mode – in monitor mode, measurements received from the sensor are applied to the main state estimate. While in this mode, a pre-update residual likelihood function is used to monitor sensor performance. The likelihood function acts as a moving window to compile a cumulative statistical likelihood for a set of residuals. Then a predefined chi-square threshold is used to determine if the set of residuals are below or above the expected distribution. If above, a failure detection will be triggered.

Validation Mode – in validation mode, sensors are excluded from having an impact on the main state estimates. Like monitor mode, ARMAS employs the likelihood function to monitor the statistical distribution of a state estimate composed of recent Kalman pre-update residuals. If ARMAS can't reinitialize the sensor, then the failed sensor gets pushed to calibration mode.

Calibration Mode – for calibration mode, the user-selectable sensor parameters for the nonlinear measurement function and the additional error states needed to process sensor measurements are estimated using residual monitoring from the trusted sensors where the observability of the system state is guaranteed. For single calibration parameter, ARMAS attempts to correct it using residual monitoring and resends it to validation mode. Linked extrinsic calibration parameters are estimated and sequenced based on the intersection of the state covariance matrix to maintain observability. If the recalibrated sensor fails to pass sensor validation, the sensor enters the remodel mode.

Remodel Mode – in remodel mode, ARMAS tries to modify the measurement model based on user-defined measurement models. Concurrent filters are used. Each filter spawned has a unique measurement model that produces measurement residuals to be compared to the core navigation estimate state. The sensor measurement model that best matches the distribution of the residual is selected and pushed into validation

mode.

Failed Mode – when remodel mode does not result in a new model selection, and Resilient Sensor Recovery (RSR) is on, the sensor gets pushed to failed mode. Periodically, the failed sensor model will re-enter validation mode based on a user-selectable time.

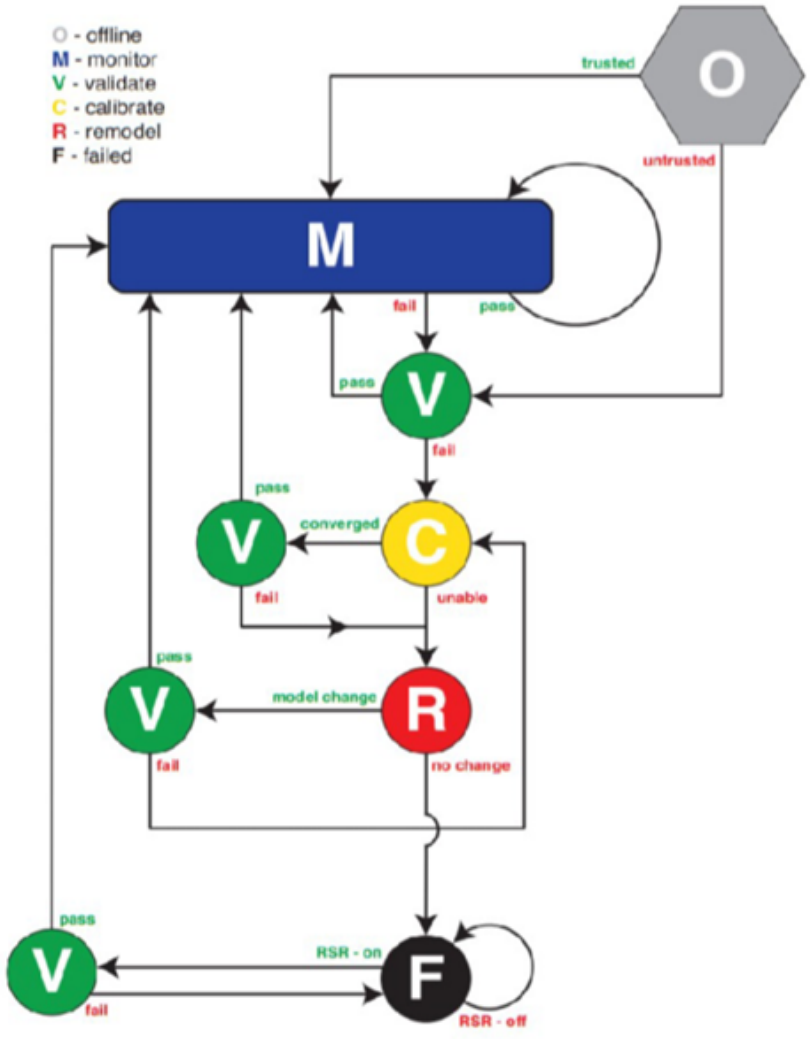


Figure 7: ARMAS framework mode transition diagram [31]

2.8 Background Summary

This chapter presents a brief technical summary of forming valid navigation so-

lutions and how INS and GPS provide data leading to the solution. This chapter also provides background on Recursive Model Estimation and how the algorithm is used to estimate the state of a system to provide a more accurate integrated navigation solution using INS and GPS data. Lastly, it summarizes related research in RAIM and ARMAS, in which both methods provide an enhanced level of integrity when using a total system to provide a navigation solution. While research has been done in each realm separately, RAIM and ARMAS have not been compared to each other in a quantitative manner to determine which method is more accurate, reliable, and effective in different scenarios. This thesis contributes to the field of navigation, specifically Position, Navigation, and Timing (PNT) [33], by providing a demonstration of RAIM and ARMAS, and a data analysis evaluating the two in different navigation scenarios.

III. Methodology

3.1 Objective

The objective of this research is to demonstrate, analyze, and compare the performance of RAIM and ARMAS systems. The results of the experiment presented in the following section serve to evaluate the two integrity monitoring systems, and test how accurate, reliable, and effective they are in realistic navigation scenarios. The experimentation attempts to specifically achieve the following objectives:

1. Design a high performance and reliable RAIM system
2. Perform a controlled experiment: validate the performance of the RAIM system developed in the thesis by using simulated data.
3. Examine the reliability and effectiveness of identifying the biased satellite in the RAIM system and the ARMAS framework.
4. Measure the accuracy of the receiver's position state in both systems during realistic reliable navigation scenarios.
5. Evaluate and compare the results provided by the two systems.

The evaluation of the results from this comparison provides quantitative data of the performance of these two systems to the PNT community, laying the foundation for the further pursuit of increasing the reliability and trustworthiness of advanced navigation systems.

3.2 Unmanned Aerial Vehicle (UAV) Flight Data

The ANT Center provided flight data is from an experiment conducted on September 30, 2021 at Camp Atterbury, Indiana. The experiment involved an UAV as the

main vehicle and a car as the secondary vehicle. Both vehicles were equipped with ranging radios and receivers to collect GNSS data. For this research, the flight data received from that day is used as the truth position, velocity, and pseudorange. Receivers for logging 8 L1 satellites' data were used during this experiment to be able remove and/or add GPS measurements for this research.

Additionally, two stationary ranging radios were placed on tripods two meters above the ground. One of the stationary radios was located at the south end of the runway and the second near the center of the runway. The radios used in this experiment are to be identified as such: UAV radio (Radio 1 ID 100), Car radio (Radio 2 ID 101), Center runway radio (Radio 3 ID 102), and South runway radio (Radio 4 ID 103). The repeated car path during the duration of the flight (blue path), the two stationary radios (green dots), and the origin location of where the UAV's flight was initiated (red rectangle) are shown in Figure 8. The origin is where the ENU coordinates frame is centered (Lat: 39.346186, Lon: -86.009495, Ht: 187.02439).

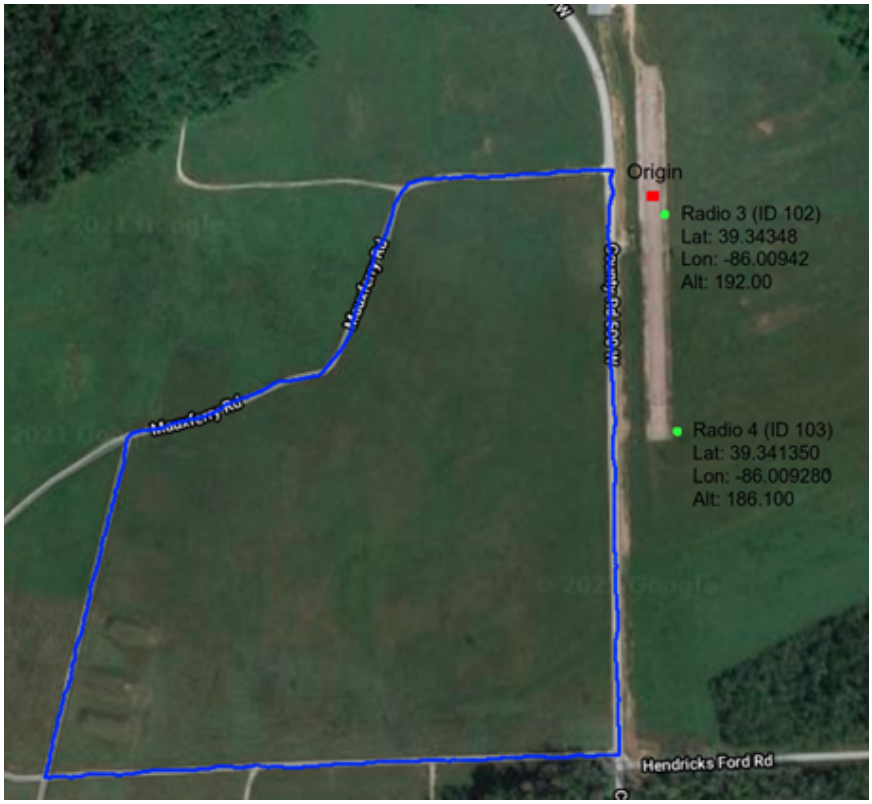


Figure 8: Top view of the runway, car path, origin, and stationary ranging radio locations

The duration of the flight test was 1277 seconds or 21 minutes and 17 seconds. The data was post-processed and down-sampled to 1 Hz, leading to 1277 truth positions, velocities, and pseudorange measurements to use in this research. However, there are 'nan' values spread throughout the data in sections where during the flight some of the satellites were not in view when collecting data. Therefore, the satellites and specific times chosen were used when coming up with the scenarios in Section 3.4, to make sure that the data received accurately represents the scenario stated.

3.3 Residual-Based Receiver Autonomous Integrity Monitoring (RBRAIM)

For decades, RBRAIM has been a reliable go-to algorithm when dealing with aerial navigation. RBRAIM is a widely popular RAIM method that is known for its ability

to rapidly respond to outliers [34]. The outliers contribute to the process of state estimation from the EKF and in the sum of squares of the psuedorange residuals used to calculate the test statistic for outlier detection. Once a deviation occurs between the predicted pseudorange and the actual measured pseudorange measurement, the test statistics of the RBRAIM detector is quickly activated. This method/algorithm entails two tests, a global and local test as shown in Figure 25 [2, 35, 36, 37]. The following subsections describe the state estimator used in RBRAIM as well as the steps and equations used in both tests.

3.3.1 EKF Modeling and Tuning

To meet the objective of calculating a predicted state and pseudorange for RBRAIM, an EKF is used. Two different EKF set ups were synthesized to determine which would be best to use in the RBRAIM system. Table 2 lists the differences between the two EKFs. The first EKF is comparable to the set up in Section 2.5 and is modeled after [38]. This EKF uses 8 states (x_i): x-position (x), x-velocity ($\dot{x}$), y-position (y), y-velocity ($\dot{y}$), z-position (z), z-velocity ($\dot{z}$), clock bias ($c\Delta t$), and clock drift ($c\dot{\Delta} t$). For the state transition matrix (F), a consistent/kinematic velocity model with the assumption that there is no coupling between $x, y, z, c\Delta t$ is created [39],

$$F_{cv} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \tag{26}$$

$$F = \begin{bmatrix} F_{cv} & 0 & 0 & 0 \\ 0 & F_{cv} & 0 & 0 \\ 0 & 0 & F_{cv} & 0 \\ 0 & 0 & 0 & F_{cv} \end{bmatrix}. \tag{27}$$

The process noise covariance matrix (Q) is a block diagonal matrix as well, but

with 3 identical blocks for the position-velocity pairs and a 2x2 submatrix for the clock bias-clock drift pair. In summary, the state components are assumed to be statistically correlated in pairs (position and velocity) but outside this pairing there is no correlation. For the position-velocity state pair, two variance terms and one covariance term is used creating an upper triangle 2x2 submatrix (Q_{xyz}) where white process noise is used to represent the uncertainty in the velocity (σ^2_{xyz}). The clock bias and clock drift have a more complex 2x2 covariance submatrix (Q_{clk}) composed of the white noise for the clock bias (S_{cb}) and the white noise for clock drift (S_{cd}). With both 2x2 submatrices, the final form of the 4x4 block process noise covariance matrix is produced:

$$Q_{xyz} = \sigma^2_{xyz} \begin{bmatrix} \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & \Delta t \end{bmatrix}, \tag{28}$$

$$Q_{clk} = \begin{bmatrix} S_{cd} + \frac{S_{cb}\Delta t^3}{3} & \frac{S_{cb}\Delta t^2}{2} \\ \frac{S_{cb}\Delta t^2}{2} & S_{cb}\Delta t \end{bmatrix}, \tag{29}$$

$$Q = \begin{bmatrix} Q_{xyz} & 0 & 0 & 0 \\ 0 & Q_{xyz} & 0 & 0 \\ 0 & 0 & Q_{xyz} & 0 \\ 0 & 0 & 0 & Q_{clk} \end{bmatrix}. \tag{30}$$

The pseudorange measurement error covariance matrix is a diagonal matrix with the same measurement error variance (σ^2_r) for the case where the number of satellites in view (n) is equal to 8. Being diagonal, means the 8 pseudorange measurements are assumed statistically uncorrelated

$$R = \begin{bmatrix} \sigma_r^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_r^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_r^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_r^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_r^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_r^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_r^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_r^2 \end{bmatrix}. \tag{31}$$

The pseudorange measurement matrix (H) is obtained by linearizing the ranging equation ($h(\hat{x}_m)$):

$$h\left(\hat{x}_{\bar{m}}\right) = \sqrt{\left((x_n - x)^2 + (y_n - y)^2 + (z_n - z)^2\right)}, \tag{32}$$

with $(x_n, y_n,\ z_n)$ being the n's satellite's coordinates, $n = 1, 2, \ldots 8$, and (x, y, z) being the receiver's coordinates, and m representing the number of states being used. By linearizing the ranging equation, it forms a matrix of partial derivatives or a Jacobian matrix of size $n \times m$. This results in the H measurement matrix used in the linear Kalman filter.

$$\frac{\partial \rho_n}{\partial x} = \frac{-(x_n - \hat{x}_{\bar{0}})}{\sqrt{((x_n - \hat{x}_{\bar{0}})^2 + (y_n - \hat{x}_{\bar{2}})^2 + (z_n - \hat{x}_{\bar{4}})^2}} \tag{33}$$

$$\frac{\partial \rho_n}{\partial y} = \frac{-(y_n - \hat{x}_{\bar{2}})}{\sqrt{((x_n - \hat{x}_{\bar{0}})^2 + (y_n - \hat{x}_{\bar{2}})^2 + (z_n - \hat{x}_{\bar{4}})^2}} \tag{34}$$

$$\frac{\partial \rho_n}{\partial z} = \frac{-(\ z_n - \hat{x}_{\bar{4}})}{\sqrt{((x_n - \hat{x}_{\bar{0}})^2 + (y_n - \hat{x}_{\bar{2}})^2 + (z_n - \hat{x}_{\bar{4}})^2}} \tag{35}$$

where n is the specific satellite, $\hat{x}_{\bar{m}}$ being the specific state used. In this case $\hat{x}_{\bar{m}} = (x, \dot{x}, y, \dot{y}, z, \dot{z}, c\Delta t, c\dot{\Delta} t)$ with the index of the state starting with 0. This leads to the H matrix final form:

$$H = \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}_{\bar{m}}} \tag{36}$$

$$H = \begin{bmatrix} \frac{\partial \rho_1}{\partial x} & 0 & \frac{\partial \rho_1}{\partial y} & 0 & \frac{\partial \rho_1}{\partial z} & 0 & 1 & 0 \\ \frac{\partial \rho_2}{\partial x} & 0 & \frac{\partial \rho_2}{\partial y} & 0 & \frac{\partial \rho_2}{\partial z} & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \rho_8}{\partial x} & 0 & \frac{\partial \rho_8}{\partial y} & 0 & \frac{\partial \rho_8}{\partial z} & 0 & 1 & 0 \end{bmatrix} \tag{37}$$

with a value of 1 in the partial derivative of $c\Delta t$ state column because the partial derivative in t of t is 1. After calculating the H matrix, the rest of the steps from Section 2.5 starting at (23) are implementable.

The second EKF synthesized herein uses similar methods as above except for adding an additional state of acceleration for each axis $(\ddot{x}, \ddot{y}, \ddot{z})$ and uses the Van Loan Method to discretize in time the matrices of the state-space system F and Q [40]. Thus, an augmented state space is used. The new Q matrix contains an acceleration bias (σ_a) and time constant (τ_a) and a clock drift error bias (σ_{cd}) and time constant (τ_{cd}) now forming a diagonal matrix of size $m \times m$ or 11x11. This results in the acceleration being modeled as a First-order Gaussian Markov process.

$$Q_a = 2 * (\frac{{\sigma_a}^2}{\tau_a}) \tag{38}$$

$$Q_{cd} = 2 * (\frac{{\sigma_{cd}}^2}{\tau_{cd}}) \tag{39}$$

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & Q_a & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & Q_a & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & Q_a & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & Q_{cd} \end{bmatrix} \tag{40}$$

Q_a represents the variance of acceleration and Q_{cd} represents the variance of the clock drift.

The state matrix (F) is built by using 2 submatrices, a position, velocity, acceleration matrix (F_{pva}) and a clock drift matrix (F_{cd}). The submatrices are composed of τ_a and τ_{cd} to form the full diagonal state matrix F of size $m \times m$

$$F_{pva} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & \frac{-1}{\tau_a} \end{bmatrix}, \tag{41}$$

$$F_{cd} = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-1}{\tau_{cd}} \end{bmatrix}, \tag{42}$$

$$F = \begin{bmatrix} F_{pva} & 0 & 0 & 0 \\ 0 & F_{pva} & 0 & 0 \\ 0 & 0 & F_{pva} & 0 \\ 0 & 0 & 0 & F_{cd} \end{bmatrix}. \tag{43}$$

From there, both the Q and F matrices are discretized in time according to the Van Loan method which involves turning Q to Q_d and turning F into Φ. The state transition (Φ) is found by taking the matrix exponential of F applied to the sampling period which in this case is Δt^2

$$\Phi = e^{F\Delta t^2}. \tag{44}$$

From there, Q_d can be obtained by forming the matrix L that is built from the F matrix and Q matrix. Then, taking the matrix exponential of L applied to T, it forms the M matrix where you will pull the upper right block (M_{12}) which is essentially the inverse of Φ applied to the Q_d. To finally obtain Q_d, the transpose of Φ must be applied to M_{12}.

$$L = \begin{bmatrix} -F & Q \\ 0 & F^T \end{bmatrix}^T \tag{45}$$

$$M = e^{L\Delta t^2} \tag{46}$$

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \tag{47}$$

$$M_{12} = \Phi^{-1} Q_d \tag{48}$$

$$Q_d = \Phi M_{12} \tag{49}$$

Once Q_d and Φ are calculated, those matrices are used instead of the F and Q in the new EKF algorithm.

	EKF Model 1	**EKF Model 2**
States $(\boldsymbol{x_i})$	8 $(x, \dot{x}, y, \dot{y}, z, \dot{z}, c\Delta t, c\dot{\Delta} t)$	11 $(x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y}, z, \dot{z}, \ddot{z}, c\Delta t, c\dot{\Delta} t)$
Process Noise Covariance $(\boldsymbol{Q})$	Non-Discretized (Q)	Discretized using Van Loan Method (Q_d)
State Matrix $(\boldsymbol{F})$	Consistent/Kinematic Velocity Model (F)	First-order Gaussian Acceleration Model (Φ)

Table 2: Differences between EKF Model 1 and 2

To determine which EKF was best to use for the RBRAIM algorithm, tests for lowest error for position, velocity, and residual were conducted using the flight data obtained with no biases added. The upper and lower bounds on the plots represent the 2-sigma state error covariance over each time step. This serves as a purpose of determining if the filter is doing well predicting the receiver position or not. A "good" filter with a 2-sigma bound should have no more than around 5% of its errors outside of the bound. The following plots and tables below show the results for both EKFs.

	$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{z}$	$\boldsymbol{\dot{x}}$	$\boldsymbol{\dot{y}}$	$\boldsymbol{\dot{z}}$
Average	0.7868	0.8448	0.8492	2.2671	2.3499	1.0923
Biggest Outlier	4.0022	7.2313	7.5746	13.1889	17.8362	15.5992

Table 3: Error Table for EKF 1

	x	y	z	$\dot{x}$	$\dot{y}$	$\dot{z}$
Average	0.6870	0.6683	0.9660	1.8200	2.0215	1.4206
Biggest Outlier	6.0820	6.2871	8.0577	16.6690	14.1748	17.3826

Table 4: Error Table for EKF 2

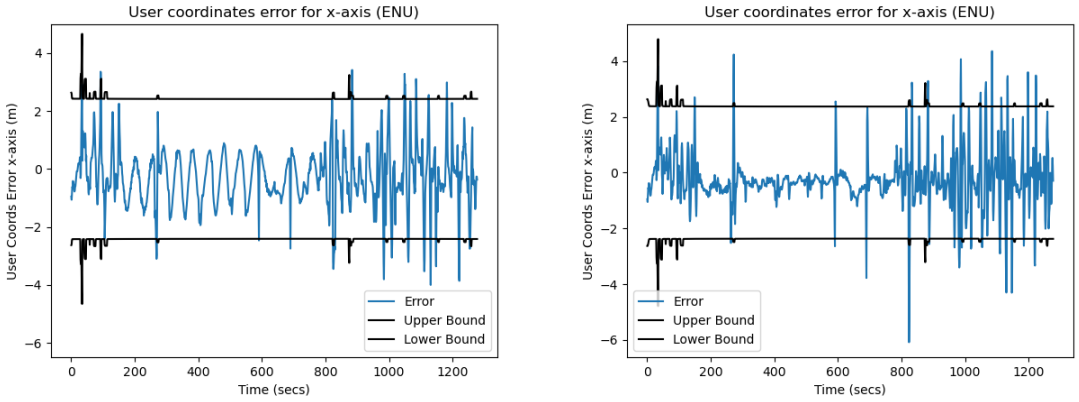


Figure 9: x-axis Position Error of EKF 1 and EKF 2 with a 2 Sigma Bound

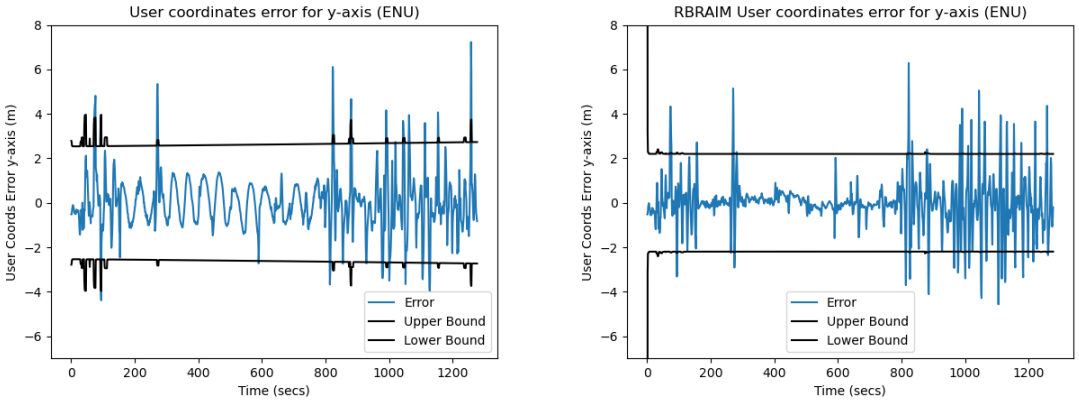


Figure 10: y-axis Position Error of EKF 1 and EKF 2 with a 2 Sigma Bound

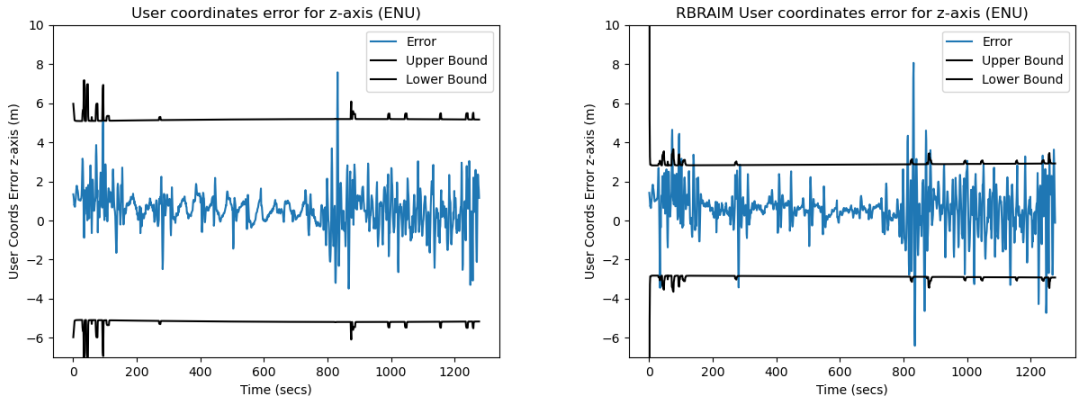


Figure 11: z-axis Position Error of EKF 1 and EKF 2 with a 2 Sigma Bound

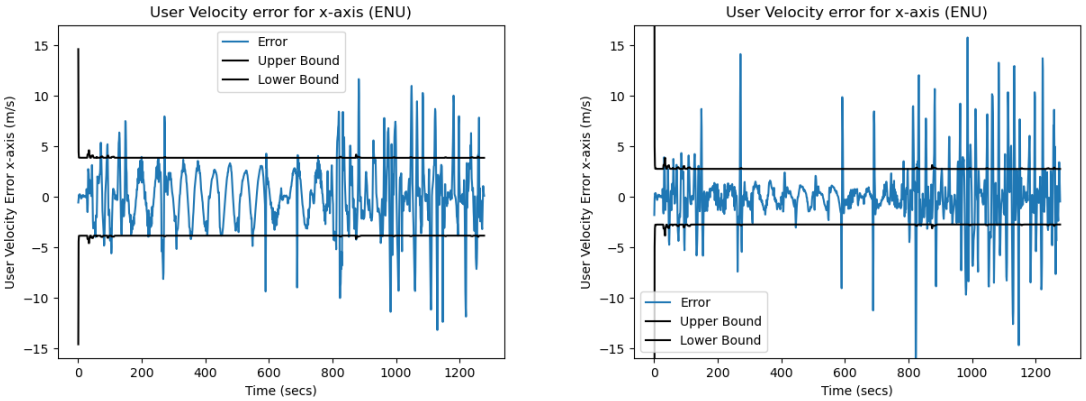


Figure 12: x-axis Velocity Error of EKF 1 and EKF 2 with a 2 Sigma Bound

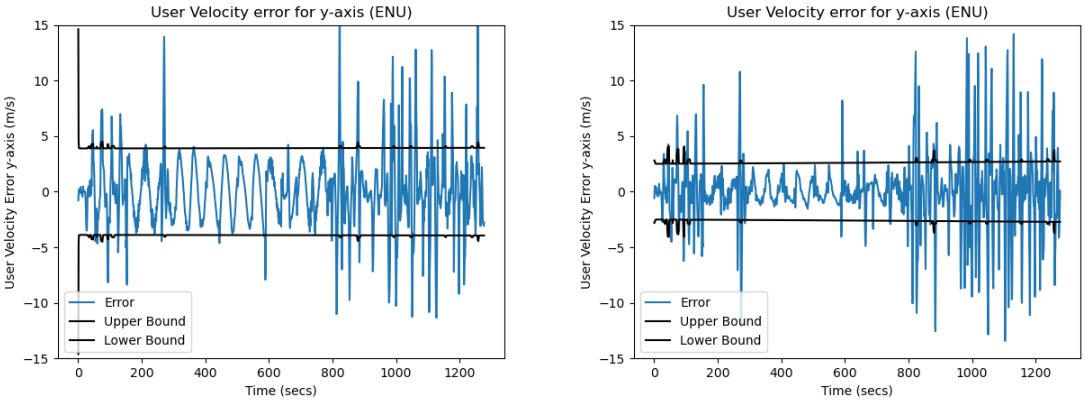


Figure 13: y-axis Velocity Error of EKF 1 and EKF 2 with a 2 Sigma Bound

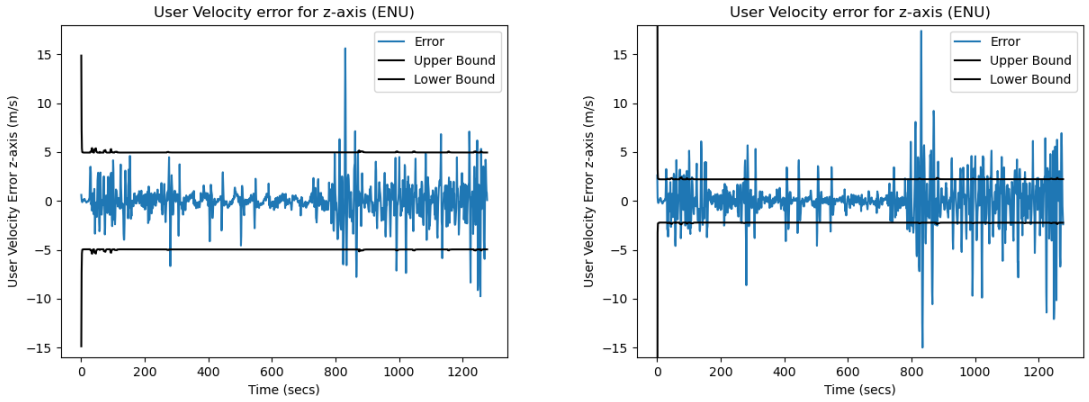


Figure 14: z-axis Velocity Error of EKF 1 and EKF 2 with a 2 Sigma Bound

	SV0	**SV1**	**SV2**	**SV3**	**SV4**	**SV5**	**SV6**	**SV7**
Average	3.948	2.275	4.056	1.776	3.762	2.741	4.091	3.970
Biggest Outlier	22.152	16.594	25.626	21.320	24.217	16.286	21.962	21.092

Table 5: Pseudorange Residual Table for EKF 1

	SV0	**SV1**	**SV2**	**SV3**	**SV4**	**SV5**	**SV6**	**SV7**
Average	2.996	1.959	3.416	1.469	2.869	2.266	3.108	3.030
Biggest Outlier	22.442	15.569	25.950	17.862	27.118	21.856	22.880	27.927

Table 6: Pseudorange Residual Table for EKF 2

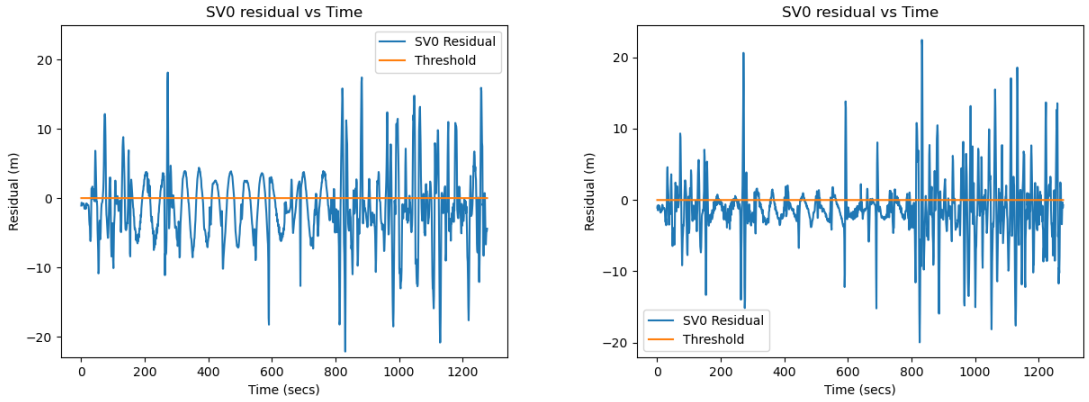


Figure 15: Pseudorange Measurement Residuals of SV0 of EKF 1 and EKF 2

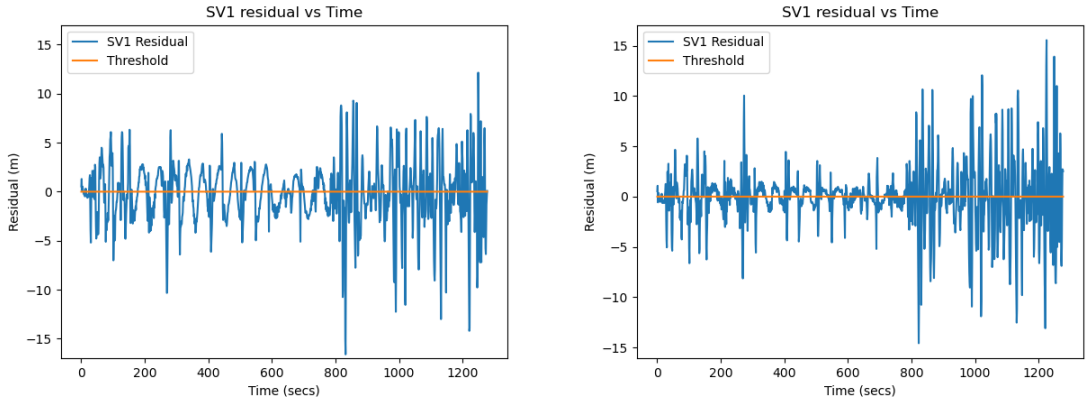


Figure 16: Pseudorange Measurement Residuals of SV1 of EKF 1 and EKF 2

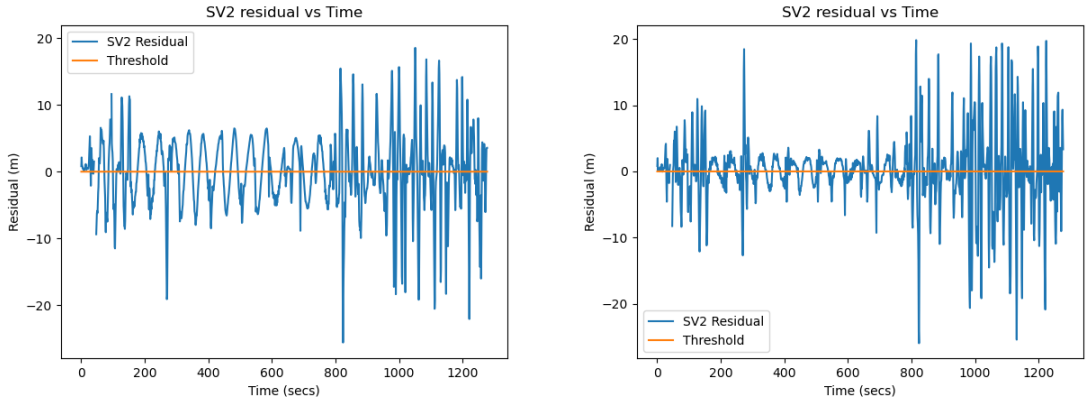


Figure 17: Pseudorange Measurement Residuals of SV2 of EKF 1 and EKF 2

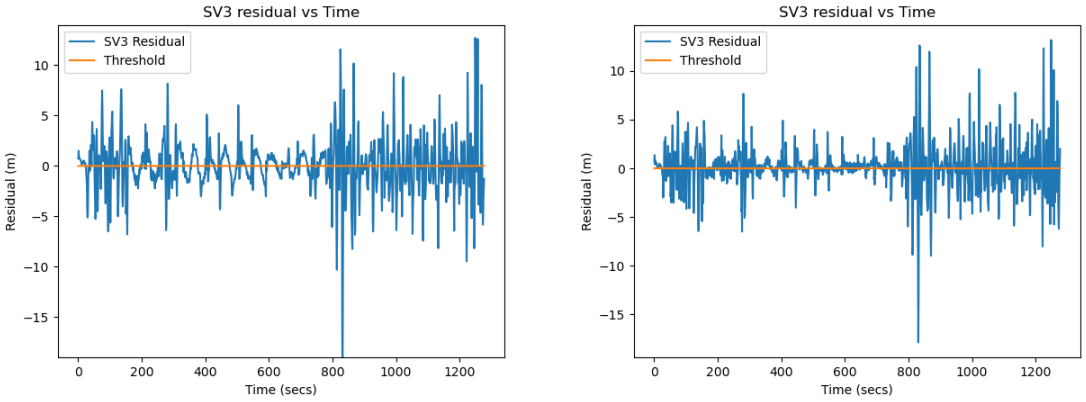


Figure 18: Pseudorange Measurement Residuals of SV3 of EKF 1 and EKF 2

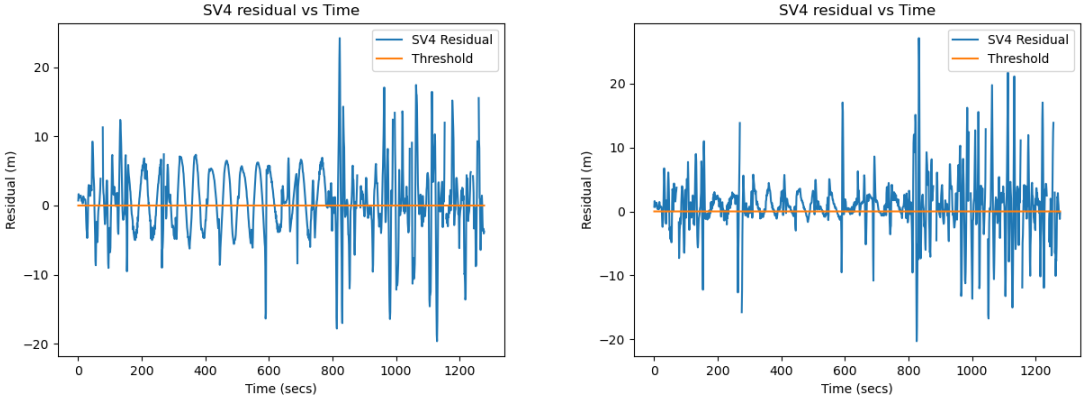


Figure 19: Pseudorange Measurement Residuals of SV4 of EKF 1 and EKF 2

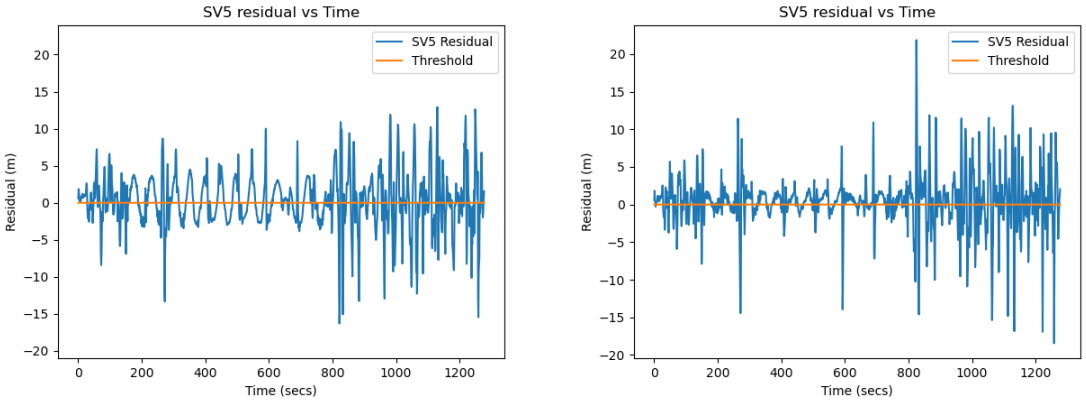


Figure 20: Pseudorange Measurement Residuals of SV5 of EKF 1 and EKF 2

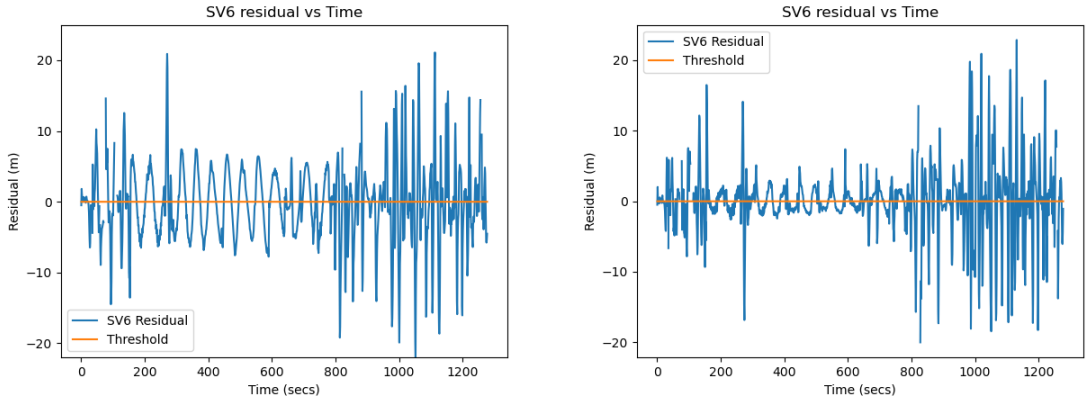


Figure 21: Pseudorange Measurement Residuals of SV6 of EKF 1 and EKF 2

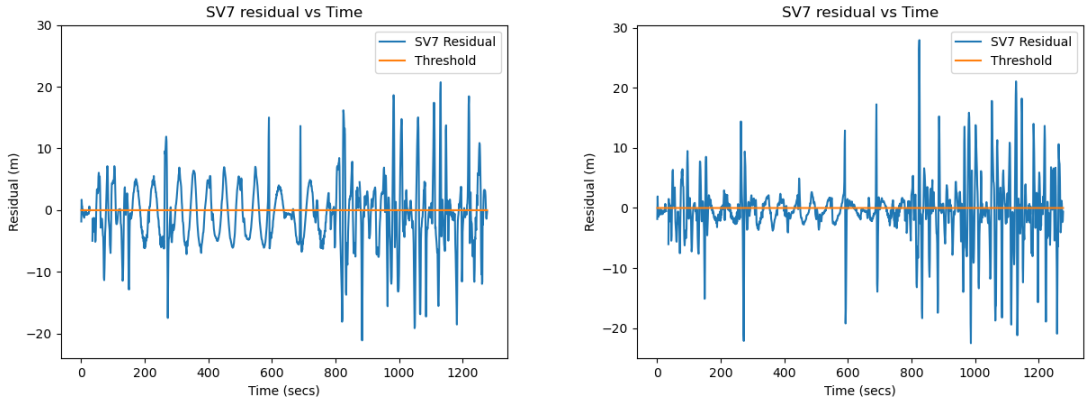


Figure 22: Pseudorange Measurement Residuals of SV7 of EKF 1 and EKF 2

After analyzing the performance of both EKFs, EKF2 was chosen due to it having a more stable and lower error metric overall when the flight data was used, compared to the performance of EKF 1.

3.3.2 RBRAIM Global Test

After building the state estimator, the process of outlier detection begins. Outlier detection starts with performing the global test to detect the existence of outliers. For each time step, the global test is used to assess the pseudorange residuals and determine if there are any errors. This is performed by taking the sum squared error of

the measurement residuals from (21) and then square rooting the solution to calculate the test statistic (t) [3]:

$$t = \sqrt{r^T r} \sim \chi^2_{n-m} \tag{50}$$

where n is the number of SVs in view and m represents the number of states to be estimated. For this RBRAIM build, $n = 8$ representing 8 satellites in view, and $m = 4$ representing the position in each axis and the clock bias due to these states affecting the pseudorange residuals. In a fault-free situation, the test statistic follows a central chi-square distribution with $(n - m)$ degrees of freedom.

The threshold of the global test (T_{RB}) can then be calculated using a central chi-sqaure distribution with the parameters degrees of freedom $(n - m)$ and possibility of a false alarm (P_{fa}) [41].

$$T_{RB} = \chi^2_{1-P_{fa},n-m} \tag{51}$$

The P_{fa} represents the probability of a non-biased navigation solution being detected as a biased solution. Therefore, $1 - P_{fa}$ becomes the percent of confidence of the detection threshold. The global threshold is used to evaluate the test statistic. If the test statistic exceeds the threshold, then the receiver is notified of an integrity failure and the local test is then performed.

3.3.3 RBRAIM Local Test

The local test is the second step in the process of outlier detection. The local test does not initiate until the global test fails (i.e., detects the existence of an outlier affecting the solution). This process is the excluding portion of the FDE function. The local test is performed specifically to identify which pseudorange measurement is

causing the inconsistency and reject it from the solution [42, 43]. This is performed by scaling the residuals:

$$C_r = HPH^T + R \tag{52}$$

$$w_i = \left| \frac{r_i}{\sqrt{(C_r)_{ii}}} \right| \quad , \; i = 1 : n, \; ii = diagonal \; of \; C_r \; matrix \tag{53}$$

The standardized residuals (w_i) are calculated by taking the absolute value of the measurement residual divided by the square root of the relevant diagonal of the residual covariance matrix (C_r) [44][45]. It is assumed that the standardized residuals are normally distributed with zero expectation in an unbiased case [36]. From there, the standardized residuals are compared to the set local threshold:

$$H_{0,i} : \; w_i \; < \; n_{1-\frac{\alpha_0}{2}} \qquad (i \; acceptable) \tag{54}$$

$$H_{a,i} : \; w_i \; \geq \; n_{1-\frac{\alpha_0}{2}} \qquad (i \; erroneous) \tag{55}$$

The null hypothesis ($H_{0,i}$) represents a situation where the i^{th} standardized residual is unbiased, and the alpha hypothesis ($H_{a,i}$) representing the case where the i^{th} standardized residual is biased. Using the probability of the local false alarm (α_0) along with the probability of the local missed detection (β_0), the possibility of a bad observation passing as a good observation, the local threshold ($n_{1-\frac{\alpha_0}{2}}$) can be found as shown in Figure 23.

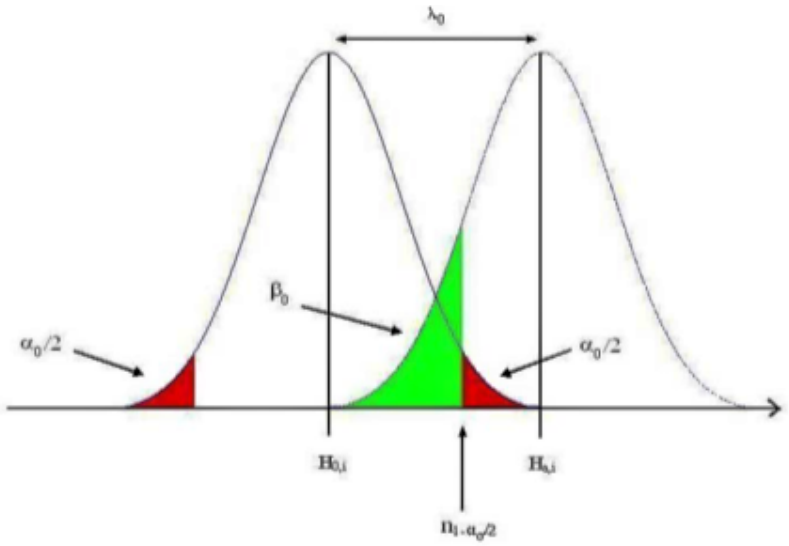


Figure 23: Normal distributions of standardized residuals for unbiased and biased cases [2]

The local noncentrality parameter (λ_0) describes the degree of difference between the biased and nonbiased standardized residual. The global test also has a specified noncentrality parameter (λ) based off its own set values of probability of false alarm (P_{fa}) and probability of missed detection (β). For the global and local test to be effective together, the parameters and probabilities must be set in a way for them to relate to each other. This relationship is dependent on if the global noncentrality parameter is the squared value of the local noncentrality parameter. Baarda's Data Snooping method [46] is used to relate the global test parameters to the local test parameters, and ultimately set the local false alarm rate based off the global false alarm rate.

$$\beta = \beta_0 \tag{56}$$

$$\lambda = {\lambda_0}^2 = \left(n_{1-\frac{\alpha_0}{2}} + n_{1-\beta_0}\right)^2 \tag{57}$$

$$\chi^2_{(\beta_0,n-m,\lambda)} = \chi^2_{(1-P_{fa},n-m)} \tag{58}$$

The global missed detection (β) is directly equivalent to the local missed detection (β_0). As stated above, the global non-centrality parameter (λ) is equal to the local non-centrality parameter (λ_0) squared. Likewise, both are equal to the total of normal distribution with $1-\frac{\alpha_0}{2}$ being the critical value and the normal distribution with $1-\beta_0$ as a critical value squared. Additionally, the chi-square distribution with β_0 as the critical value, $n-m$ as the number of degrees of freedom, and λ as the location is equivalent to the chi-square distribution with $1-P_{fa}$ as the critical value and $n-m$ as the number of degrees of freedom. With these equations, the local false alarm probability can be obtained when setting the global false alarm and missed detection probabilities.

The measurement having the largest standardized residual is regarded as the outlier and removed from the navigation solution: meaning, the k^{th} measurement is erroneous when:

$$w_k \;\geq\; w_i \;\forall i \quad \wedge \quad w_k \;\geq\; n_{1-\frac{\alpha}{2}} \tag{59}$$

The standardized residual suspected to be erroneous (w_k) must be greater than or equal to all the other standardized residuals and greater than or equal to the threshold. This method of single fault isolation is known as the Maximum Residual Algorithm. This scheme assumes that only ONE satellite will be faulty per timestep. Therefore, to detect multiple blunders, the Maximum Residual Algorithm can be run recursively. Once an outlier is found and excluded, the test is repeated on the remaining subsample after the previous outlier is removed as shown in Figure 24 [47].

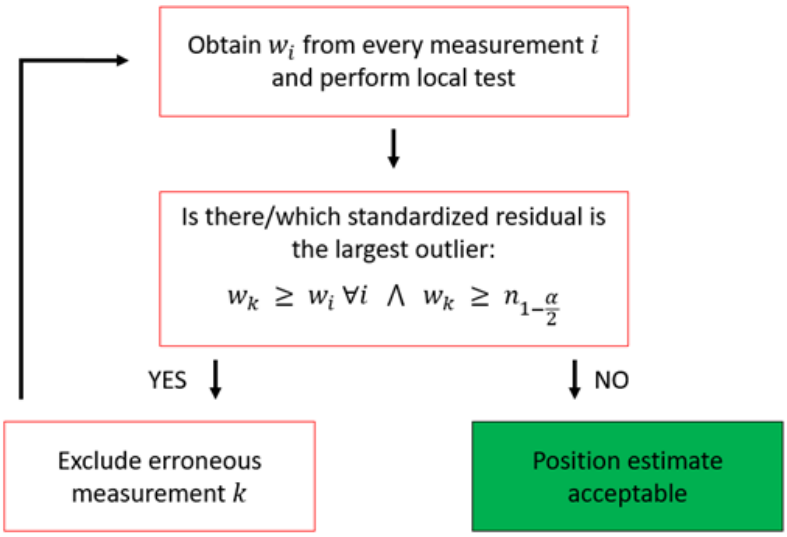


Figure 24: Local Sequential Test Flow Chart

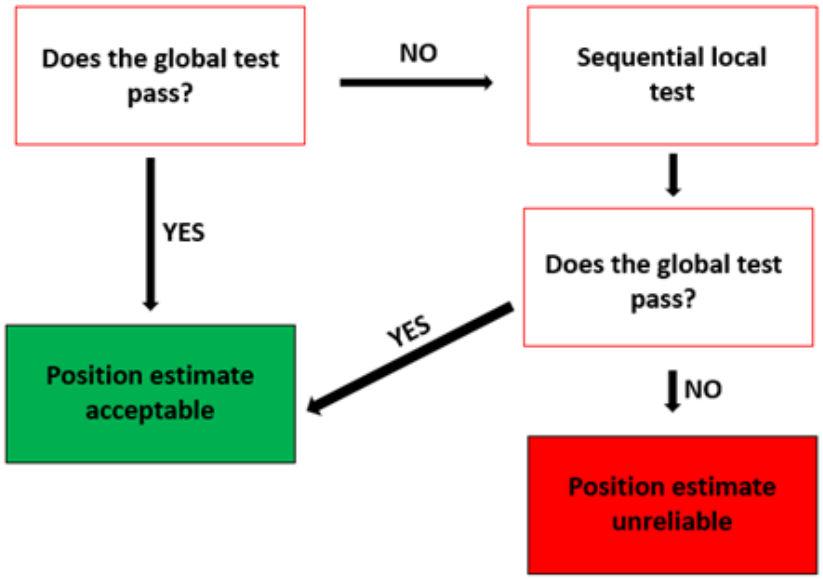


Figure 25: FDE scheme of RBRAIM based on global and sequential local test

The parameter values used for both the local and global test are mentioned in Table 7 of the next section. The values chosen for each parameter was calculated based off of ARMAS's set Global Probability of False Alarm (1/85000) used in the experiments below. For a more detailed explanation of ARMAS's parameters and

initialization, reference Brandon Blakely's, Staff Engineer for the ANT Center, thesis "Evaluation of the ARMAS-SOM Framework with Real Data."

3.4 Scenario/Experiment Design

Using simulated data and the flight data obtained from Camp Atterbury, several scenarios will be examined in this work. The following subsections outline each of the five experiments performed to produce the results in the following chapter. See Table 8 for a listing of experiments and a brief description of each.

The first experiment is used to validate the RBRAIM built by gauging its performance against simulated data and verifying the overall functionality of the system. The following 4 experiments compare the RBRAIM to ARMAS allowing a functionality and quantitative comparison between the two systems. Each of the experiments fulfill objectives 2, 3, and 4 in Section 3.1:

Objective 2 – is accomplished by the first experiment run. The RBRAIM system built will run with the simulated data being inputted into the system. An observation of functionality will be performed to prove that the RBRAIM system is operationally sound.

Objective 3 – is accomplished by performing the last 4 experiments. The RBRAIM system and ARMAS framework uses the Camp Atterbury data for the 4 experiments. Both systems are then examined to see which system caught and excluded the biased/jammed satellite efficiently and effectively.

Objective 4 – is accomplished by the last 4 experiments as well. Both systems are compared in accuracy using the positional state error in ENU coordinates. To show a better representation of the state error, the Mean Square Error (MSE) of each time step is observed and then the Root Mean Square Error (RMSE) is obtained.

Each experiment is performed for a scenario with the satellites either being jammed

(satellite not available) or biased (fed wrong information) over a specific time period. For both systems, the acceleration bias (σ_a), acceleration time constant (τ_a), clock drift error bias (σ_{cd}), time constant (τ_{cd}), the pseudorange measurement error variance (σ_r^2), and global/local false alarm value (P_{fa}/α_0) will remain constant throughout the experiments. The ARMAS system uses the same set of variables to create the Q and R matrices as well, but only uses 10 states by dropping the clock drift state. Due to the reduction in the number of states, the variables for clock drift are moved to clock bias for the ARMAS framework. The same method is applied to the initial state covariance (P_0). Another difference between the two systems is the local test aspect of the system. While RBRAIM uses a sequential outlier method, ARMAS has a 6-mode method of checking integrity of the information provided with sensors as discussed in Section 2.7. The table below lists what each parameter/variable is set to.

	RBRAIM	ARMAS
Acceleration Bias (σ_a)	100	100
Acceleration Time Constant (τ_a)	100	100
Clock Drift/Clock Bias (σ_{cd})	8000	8000
Clock Drift/Clock Time Constant (τ_{cd})	3600	3600
Measurement Error Variance (σ_r^2)	2	2
Initial State Covariance (P_0)	Diagonal 11x11 matrix of [100 , 1, 10000, 100, 1, 10000, 100, 1, 10000, 1, 6.4x10^7]	Diagonal 10x10 matrix of [100, 1, 10000, 100, 1, 10000, 100, 1, 10000, 6.4x10^7]
Global False Alarm (P_{fa})	1/85000	1/85000
Local False Alarm (α_0)	1.93x10^{-5}	N/A (6-mode method)
Global Missed Detection (β)	0.19	N/A (6-mode method)
Local Missed Detection (β_0)	0.19	N/A (6-mode method)

Table 7: Variables' Setting/Initializing for RBRAIM and ARMAS

3.4.1 Experiment 1: Validation Test

This experiment is used to assess and test the functionality of the RBRAIM method. The simulated data is broken up by satellite data produced and receiver data produced.

The position of the satellites were obtained on 22nd of January 2022 at 14:44:07 EST. The positions pulled were from 8 stationary satellites: GPS1, GPS3, GPS6,

GPS12, GPS13, GPS15, GPS17, GPS19. To simulate the satellites orbiting, random number generators were used with the means being an average velocity of GPS3 from the Camp Atterbury data with a standard deviations of 1.0. Applying the average velocities to the 8 stationary satellite locations, produced a different position of the satellites for each time-step.

The receiver data used is from a simulated fixed wing class 2 UAV flying narrow figure eights as its flight path. The receiver data produced position and velocity in 3-dimensions for 401 time steps at a rate of one data point per second ($\Delta t = 1.0$).

With both the truth satellite data and truth receiver data, the pseudorange measurement was produced with the same clock bias (-123163.8413) as the Camp Atterbury data.

With the simulated data produced, the bias can be applied to validate the RBRAIM system. A bias of 50 meters is added to the pseudorange measurements of satellite GPS1 at the 100th timestep and GPS3 at the 200th timestep. The 50-meter bias is repeatedly added to each pseudorange measurement for 20 seconds and then ceases. Meaning the bias for GPS1 starts on the 100th timestep and ends on the 120th timestep. From there, the biased data is inputted into the RBRAIM system which yielded the results presented in the next chapter.

3.4.2 Experiment 2: Jamming Scenario 1

This experiment tests to see how well RBRAIM and ARMAS handles GPS spoofing. The Camp Atterbury data is used in this and the following experiments. In this jamming scenario, both systems will start with 8 L1 satellites in view. From there, two satellites (GPS4 & GPS26) are jammed at the same time of 700 seconds into the flight to simulate simultaneous jamming. The jamming affect lasts from the 700th timestep to the end of the recorded flight at the 1277th time step. The pseudorange

measurements are manipulated similarly to experiment 1, but a 'nan' value is added into the data instead of a bias. Adding a 'nan' value to the data simulates the effect of data not being available for the specified satellites. This experiment is used to determine how accurate the position data produced from RBRAIM and ARMAS is while multiple satellites are being affected.

3.4.3 Experiment 3: Jamming Scenario 2

This experiment is similar to Experiment 2. Both systems begin with 8 L1 satellites in view. Then, the first satellite (GPS22) gets jammed at the 400th time step. After another 200 time steps pass, a second satellite (GPS27) gets jammed and so on. This pattern continues for two more satellites (GPS31 & GPS9) and lasts the rest of the recorded flight. This experiment, like Experiment 2, tests the accuracy of the position data produced by both systems, and also allows an observation of accuracy drop off with the reduction in satellites.

3.4.4 Experiment 4: Biasing Scenario 1

This experiment tests to see how well RBRAIM and ARMAS handle the simultaneous biasing of the Camp Atterbury flight data. Both systems begin with 8 L1 satellites in view with no bias added. At the 600th timestep, both satellites (GPS4 & GPS26) receive a 50-meter steady bias during 150 time steps. After the 150 timesteps, the bias is removed from both satellites and both systems continue to operate on the rest of the data. This experiment is used to not only test the accuracy of the results produced by both systems, but also tests the efficiency of both systems in identifying, removing, and recovering from the bias. Most integrity systems do not handle simultaneous biasing well, so this is a way to determine how resilient the herein investigated systems are in comparison to each other.

3.4.5 Experiment 5: Biasing Scenario 2

Experiment 5 is used to assess the FDE performance of both systems. Like the previous tests, both systems begin with 8 L1 satellites in view. The first satellite (GPS3) gets biased at the 300 second mark with a ramp bias applied during 100 time steps. The ramp bias starts at 10 meters and an additional 10 meters get added to the bias every time step. Therefore, at the 100th timestep of biasing, the applied bias will be 1000 meters. After the 100th biased time step, the bias ends for GPS3 and the next satellite (GPS4) will begin with a bias of 10 meters. This process is repeated for two more satellites (GPS16 & GPS26) until the 700th time step in the flight data, and the biasing ends allowing both systems to run through the rest of the unbiased data. This experiment is used similarly to Experiment 4 for testing the efficiency of both systems FDE function and the accuracy of the data produced by both systems. However, this experiment emphasizes the detection rate by using the ramp bias as a means of determining how big the bias will become before the systems identify there is a bias.

	Exp. 1	**Exp. 2**	**Exp. 3**	**Exp. 4**	**Exp. 5**
Data Used	Simulated	Camp Atterbury	Camp Atterbury	Camp Atterbury	Camp Atterbury
Jam or Bias	Bias	Jam	Jam	Bias	Bias
# of Sats Effected	2	2	4	2	4
Which Sats	1, 3	4, 26	22, 27, 31, 9	4, 26	3, 4, 16, 26
Start of Effect (timestep)	100, 200	700, 700	400, 600, 800, 1000	600, 600	300, 400, 500, 600
Duration of Effect (s)	20 each	End of data	End of data	150 each	100 each
Bias Value Applied (m)	50	N/A	N/A	50	Ramp Bias of 10

Table 8: Summary of Experiment 1-5

3.5 RBRAIM and ARMAS Baseline

To be able to correctly interpret the results produced by the experiments in the next chapter, the baselines used in both systems must be known. The positional state error, MSE, and RMSE of RBRAIM in the simulations is provided below.

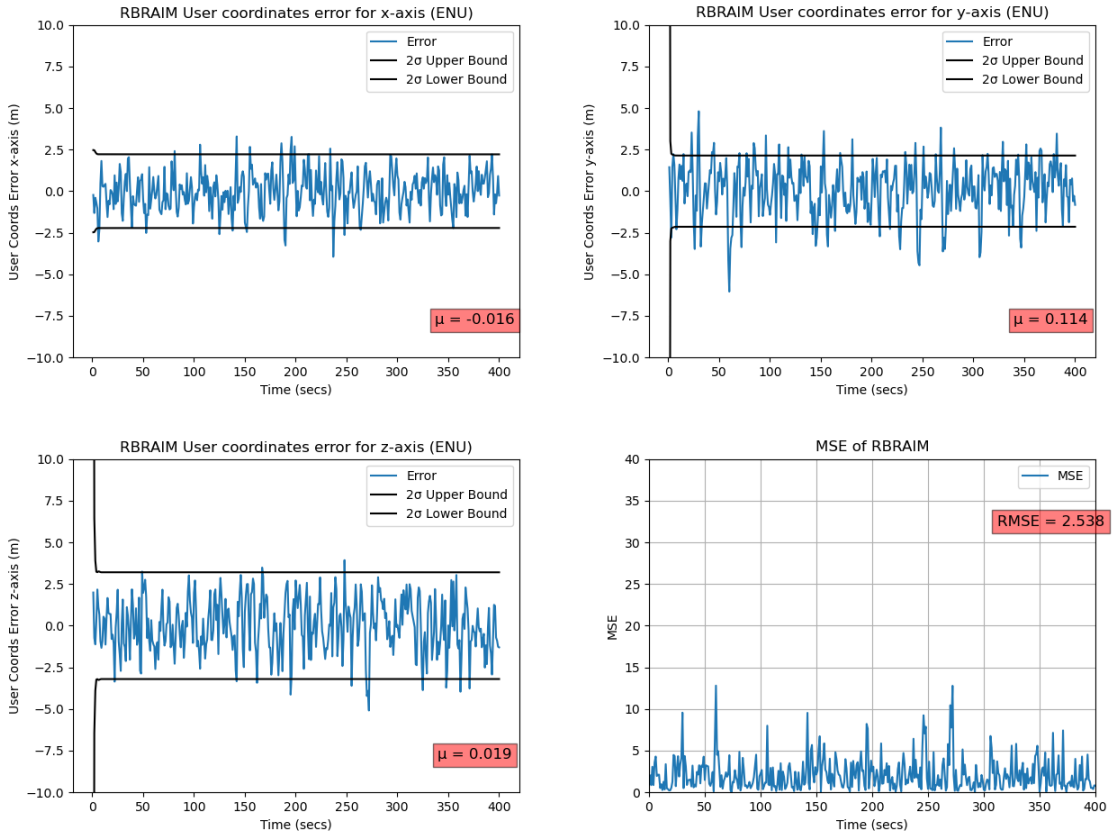


Figure 26: RBRAIM 3D Error, and MSE Plot for the Simulated Data

As mentioned in Section 3.4, the velocity added to the stationary satellites are randomly generated with a standard deviation of 1.0 of the average velocity in each axis. Therefore, the GPS positions are not the same for each time the test is run – think of a Monte Carlo (MC) experiment. The data above is not repeatable. However, after running the simulated data 20 times, the data above provides the average of the 20 runs. Thus, if the RBRAIM produces values similar to the data above, it is functioning properly.

The positional state error, MSE, and RMSE of RBRAIM and ARMAS for the Camp Atterbury data is provided below. The ARMAS results below and in the next chapter were provided by Brandon Blakely, Staff Engineer for the ANT Center.

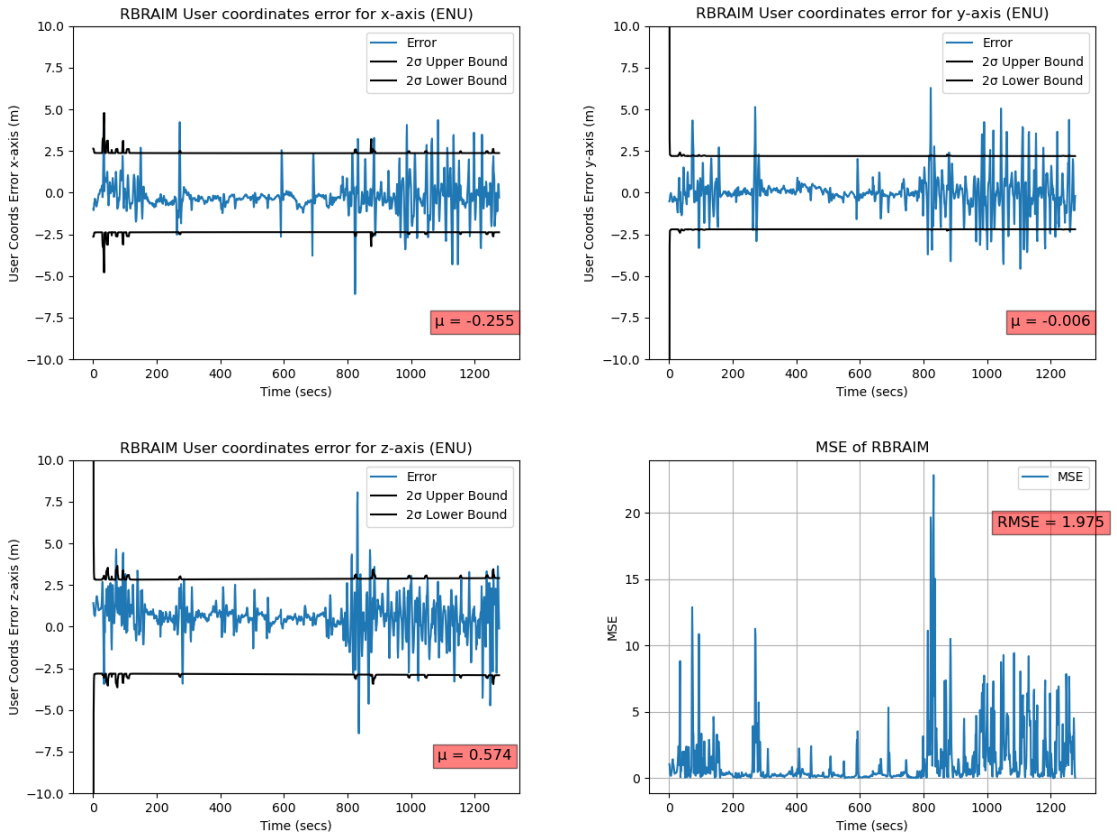


Figure 27: RBRAIM 3D Error, and MSE Plot for the Camp Atterbury Data

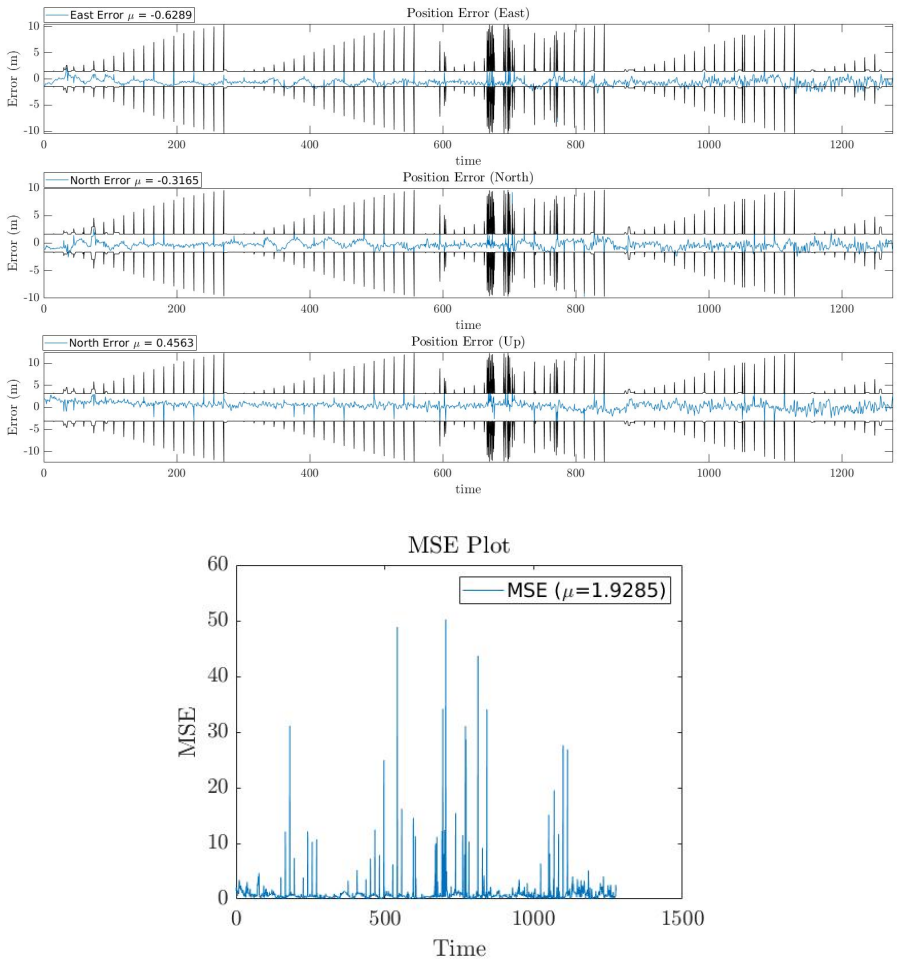


Figure 28: ARMAS 3D Error, and MSE Plot for the Camp Atterbury Data

From the information provided above, RBRAIM and ARMAS have a similar RMSE value which determines how accurate the position provided from each system is. The results in the next section will be obtained by taking the differences between the RMSE of each experiment and the baseline to then be compared to the systems. This will determine how accurate and functional both systems are and overall determine which system performed better.

IV. Results and Analysis

4.1 Overview

This chapter describes the results obtained by running the 5 experiments discussed in Chapter 3. The results of each experiment are discussed in two manners: (i) the accuracy of the systems and (ii) the functionality of the systems. The accuracy of both systems will be determined by examining the error in the state/position estimates according to the MSE plots/records provided. The systems' performance is compared to their baseline performance discussed in Section 3.5, to determine quantitatively how accurate the position provided by the systems are while being jammed and biased. Aside from Experiment 1, the systems are then compared to each other to determine which is more resilient in each scenario.

The functionality of the two systems is determined by the results of the experiments when the pseudorange measurements are biased, Experiment 1, 4, and 5. The amount of time it took for each system to flag the bias will be provided to determine how well the identification and exclusion function of each system works. For the last experiment, Experiment 5, how high the bias gets before the attendant pseudorange measurement is rejected, will also be provided.

4.2 RBRAIM and ARMAS Accuracy

In this section, the accuracy of the systems during each biasing and jamming scenario is discussed. The state error, MSE, and RMSE metrics introduced in Section 3.4 are used. The results of all five experiments are used and the performance metrics are calculated using a python script.

The state error (Δx_i) script takes the truth position state (x_i) of the receiver and subtracts it from the post-updated predicted position state ($\hat{x}_i$) over each time step

to then be plotted.

$$\Delta x_i = x_i - \hat{x}_i \tag{60}$$

In the state error plots, the 2-sigma covariance upper and lower bounds are also shown. Meaning, that if the system is working properly, 95% of the state error should be inside the bounds. When the bounds increase, it signals the EKF lost confidence in the predicted position state. When the state error is significantly outside of the bounds, it signifies an unmodeled error or more noise in the measurements than acknowledged in the design of the EKF. The script written to produce the bounds uses the diagonal elements of the predicted error covariance (P_i), the variance of the predicted state, and plots them over time.

$$2\sigma \; cov \; upper \; bound = 2 * \sqrt{(P_i)_{ii}} \quad i = current \; timestep, \; ii = diagonal \; of \; P_i \; matrix \tag{61}$$

$$2\sigma \; cov \; lower \; bound = -2 * \sqrt{(P_i)_{ii}} \quad i = current \; timestep, \; ii = diagonal \; of \; P_i \; matrix \tag{62}$$

The red, dotted lines in the position state error plots represent where each bias/-jamming starts as referenced in Section 3.4.

The MSE script calculates the MSE of the position state error by squaring each axis (x, y, z) error, summing them, and then dividing by number of states, 3, for each time step. Then, that array is plotted over time to produce an MSE plot. The MSE represents how well the system is estimating the receiver's position for each time-step.

$$MSE_i = \frac{1}{3}(\Delta {x_{x_i}}^2 + \Delta {x_{y_i}}^2 + \Delta {x_{z_i}}^2) \tag{63}$$

The MSE plot also has red, dotted lines, like the position state error plot, to represent when the bias/jamming begins.

The RMSE of the state error script is similar to the MSE script. However, instead of dividing by 3 for each time step, all of the MSE values are summed and divided by the number of time steps (n). Lastly, the value is square rooted to produce the RMSE value over the entirety of the data. The RMSE value represents how well the system is estimating the receiver's position throughout the entirety of the data. When the RMSE is above the value of 8, the system is producing poor estimations.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\Delta {x_{x_i}}^2 + \Delta {x_{y_i}}^2 + \Delta {x_{z_i}}^2)} \tag{64}$$

The plots in Figure 29, 30, 31, 32, and 33 show the results from Experiment 1-5 that the RBRAIM system provided. Figure 34, 35, 36, and 37 shows the plotted results from Experiment 2-5 that the ARMAS system provided. Table 9 summarize the results of each experiment from the systems. Table 10 displays the difference of the baseline values compared to the experiment provided values in Table 9 so that the systems can be compared to each other to determine which was more accurate in each experiment.

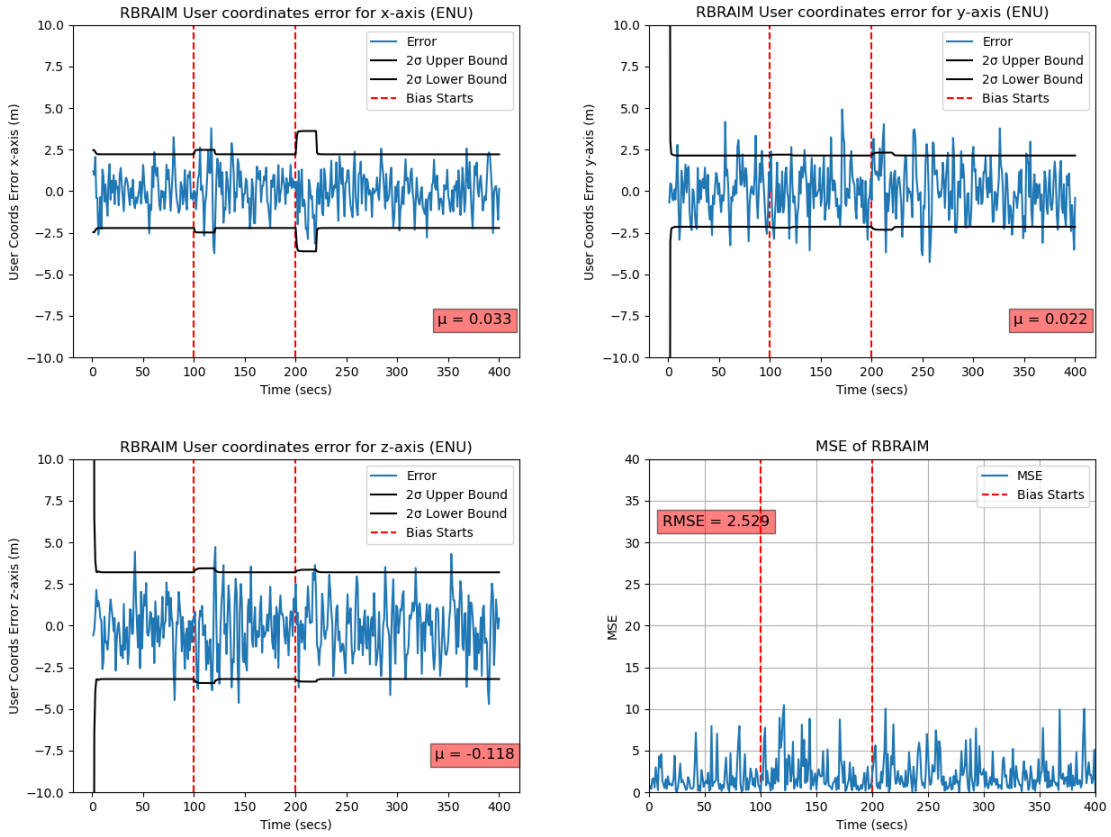


Figure 29: RBRAIM's 3D Error and MSE Plot for Experiment 1 (RBRAIM Validation Experiment)

A bias of 50 meters is added to Satellite 1 at time-step 100 and Satellite 3 at time-step 200. Both of the biases last for 20 seconds before the pseudorange goes back to normal. The covariance bound spikes at the bias implementation timesteps signifying the loss in confidence of the system outputting the state estimation, and the system does not gain the confidence back until the bias ceases 20 timesteps later. Even with the loss of confidence, the state error and MSE remained stable, and the RMSE value remained similar to the baseline's RMSE value. These results show that RBRAIM has no issue handling this scenario, resulting in a functional and valid system.

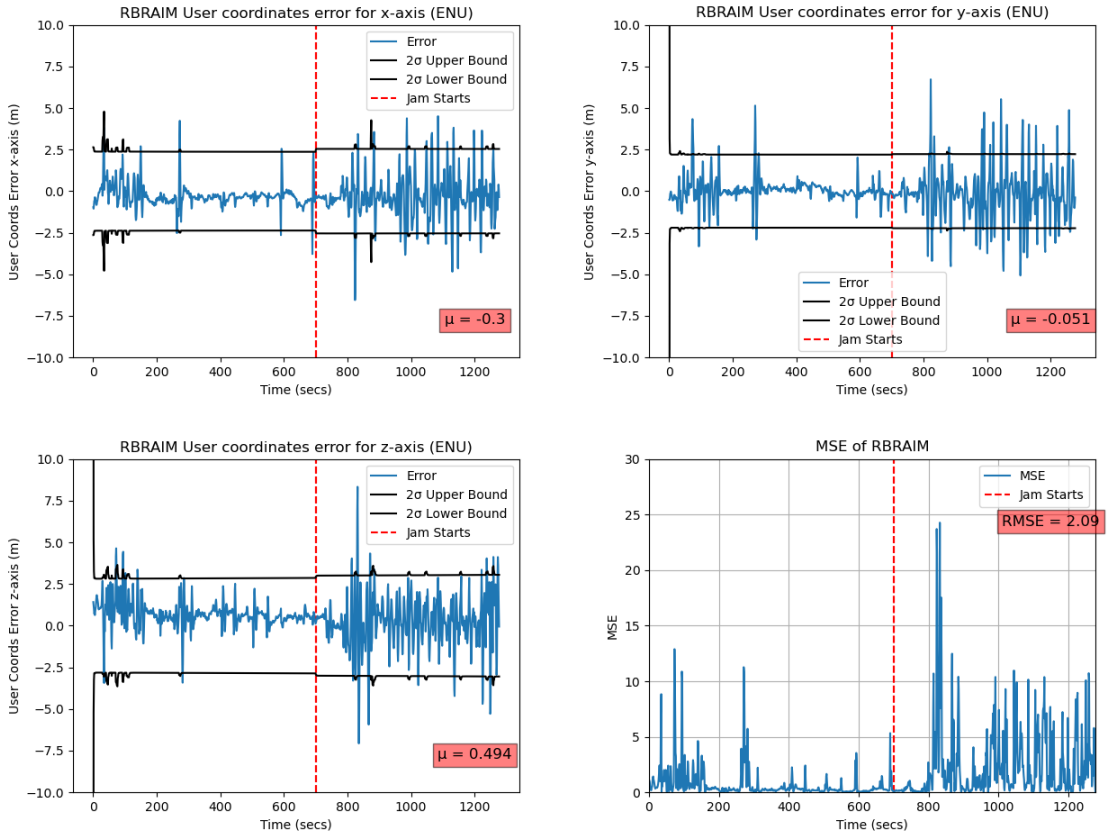


Figure 30: RBRAIM's 3D Error and MSE Plot for Experiment 2 (Simultaneous Jamming)

Satellite 4 and 26 drop out at time-step 700 and remain jammed until the end of the data. The covariance bound spikes slightly at the jamming implementation timestep, but not enough to have any significant affect on the estimation. The state error and MSE remained stable, resulting in an insignificant change of 0.115 (Table 10) between the Baseline RMSE value and Experiment 2's RMSE value. These results show that RBRAIM performed well and the simultaneous jamming had no affect on the system.

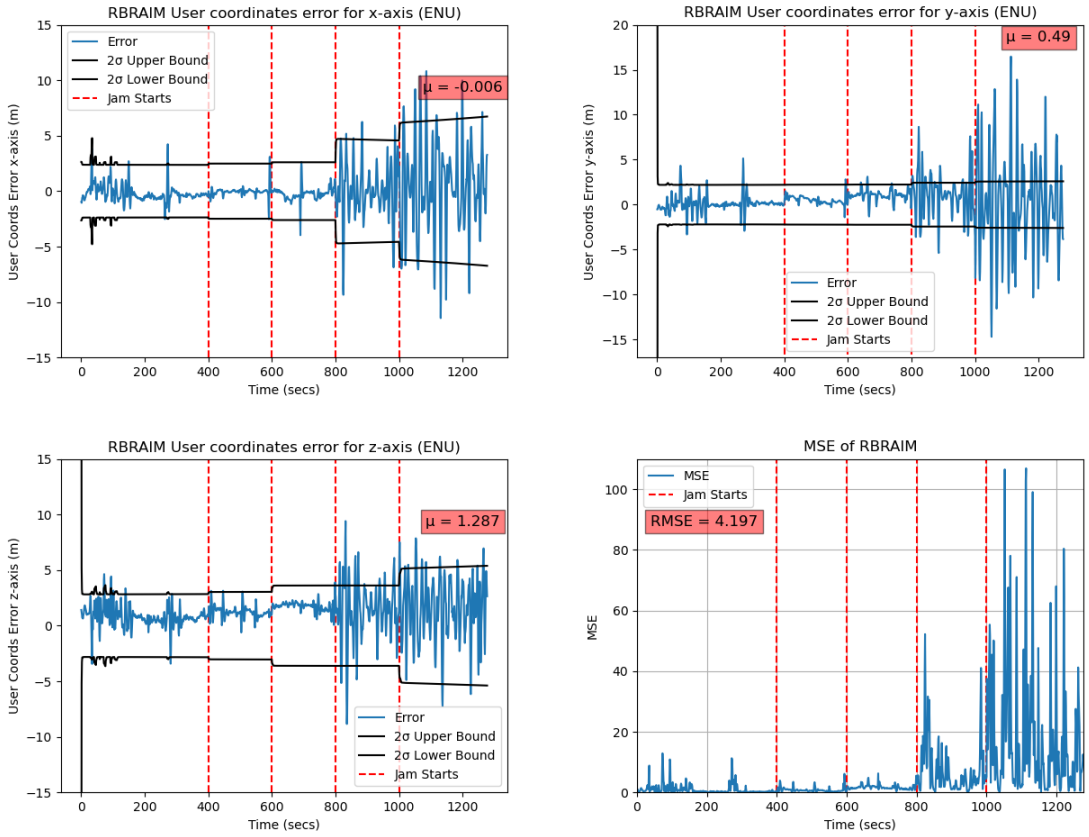


Figure 31: RBRAIM's 3D Error and MSE Plot for Experiment 3 (Periodic Jamming)

For the periodic jamming scenario, Satellite 22, 27, 31, and 9 are affected at different time-steps and remain affected until the end of the data: Satellite 22 affected at time-step 400, Satellite 27 affected at time-step 600, Satellite 31 affected at time-step 800, and Satellite 9 affected at time-step 1000. The covariance bound wasn't affected much by the first two satellites dropping out. However, when the third and last satellite dropped out, it had a noticeable affect on the systems confidence in estimation. The state error and MSE were also heavily affected when the third and last satellite got jammed resulting in the instability in the state error plots and the large values in the MSE plot after time-step 800. Due to RBRAIM performing poorly only after time-step 800, the RMSE value was still low with a value of 4.197. Therefore, RBRAIM handled the periodic jamming scenario well overall.

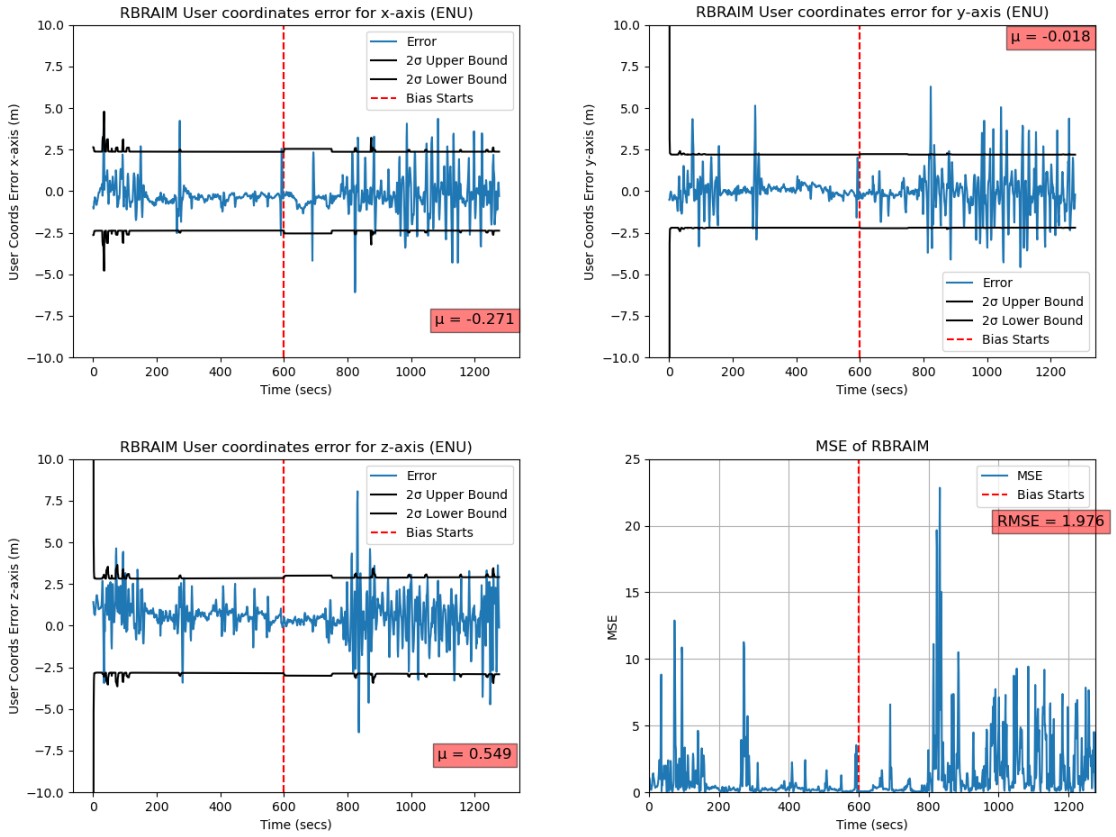


Figure 32: RBRAIM's 3D Error and MSE Plot for Experiment 4 (Simultaneous Bias)

A bias of 50 meters is added to Satellite 4 and Satellite 26 at time-step 600. Both of the biases last for 150 seconds before the pseudorange goes back to normal. The covariance bound spikes slightly at time-step 600, but not enough to have an affect on the state estimation. The state error and MSE remained stable throughout the entirety of this experiment, and the RMSE value remained statistically the same compared to the Baseline RMSE. These results show that RBRAIM essentially was unaffected by the simultaneous bias, and handling the scenario very well.

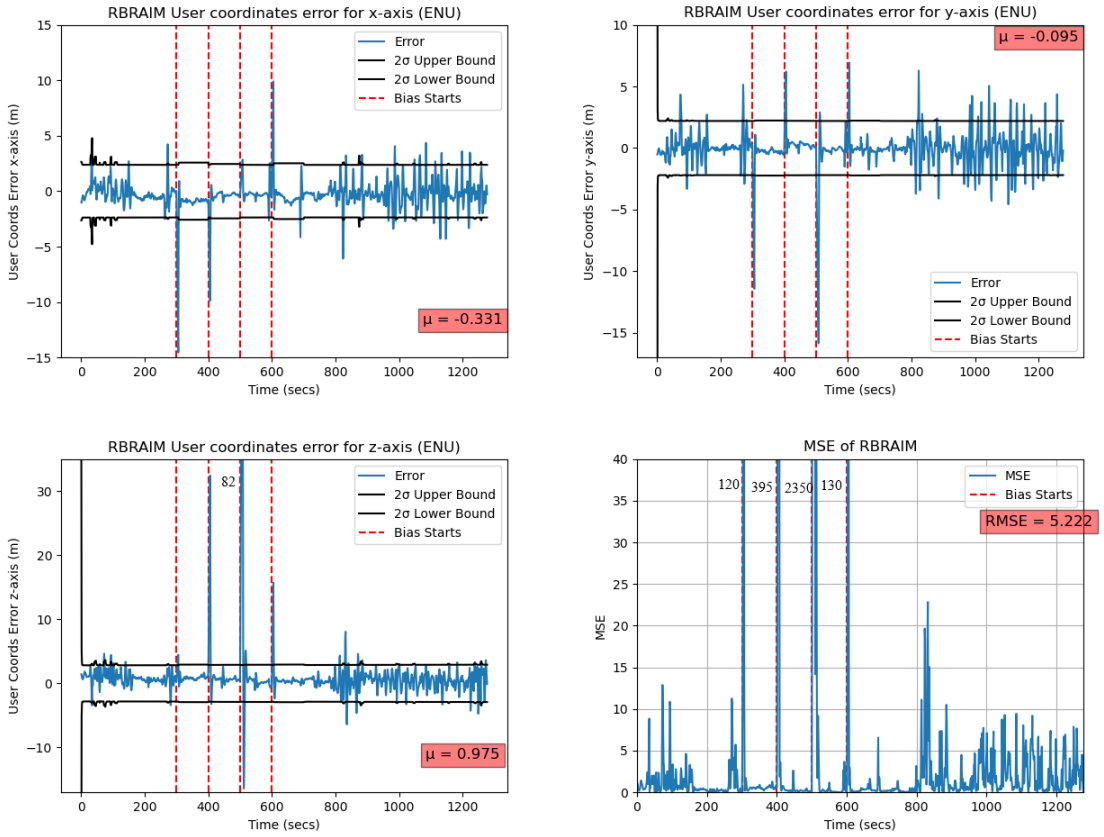


Figure 33: RBRAIM's 3D Error and MSE Plot for Experiment 5 (Periodic Ramp Bias)

A ramp bias of 10 meters is added to Satellite 3, 4, 16, and 26 at varying time-steps: Satellite 3 affected at time-step 300, Satellite 4 affected at time-step 400, Satellite 16 affected at time-step 500, and Satellite 26 affected at time-step 600. The biases last for 100 seconds before the pseudorange returns to normal and the next bias is applied. RBRAIM had a hard time dealing with this scenario. Even though the covariance bounds were not affected, the state error and MSE was. The state error and MSE spiked dramatically during each bias implementation for about 6 seconds. However, once the bias was removed, both plots returned to being stable. Even though RBRAIM performed poorly compared to the previous experiments the RMSE still remained below 8 with a value of 5.222. Meaning, RBRAIM handled the

periodic jamming scenario well.

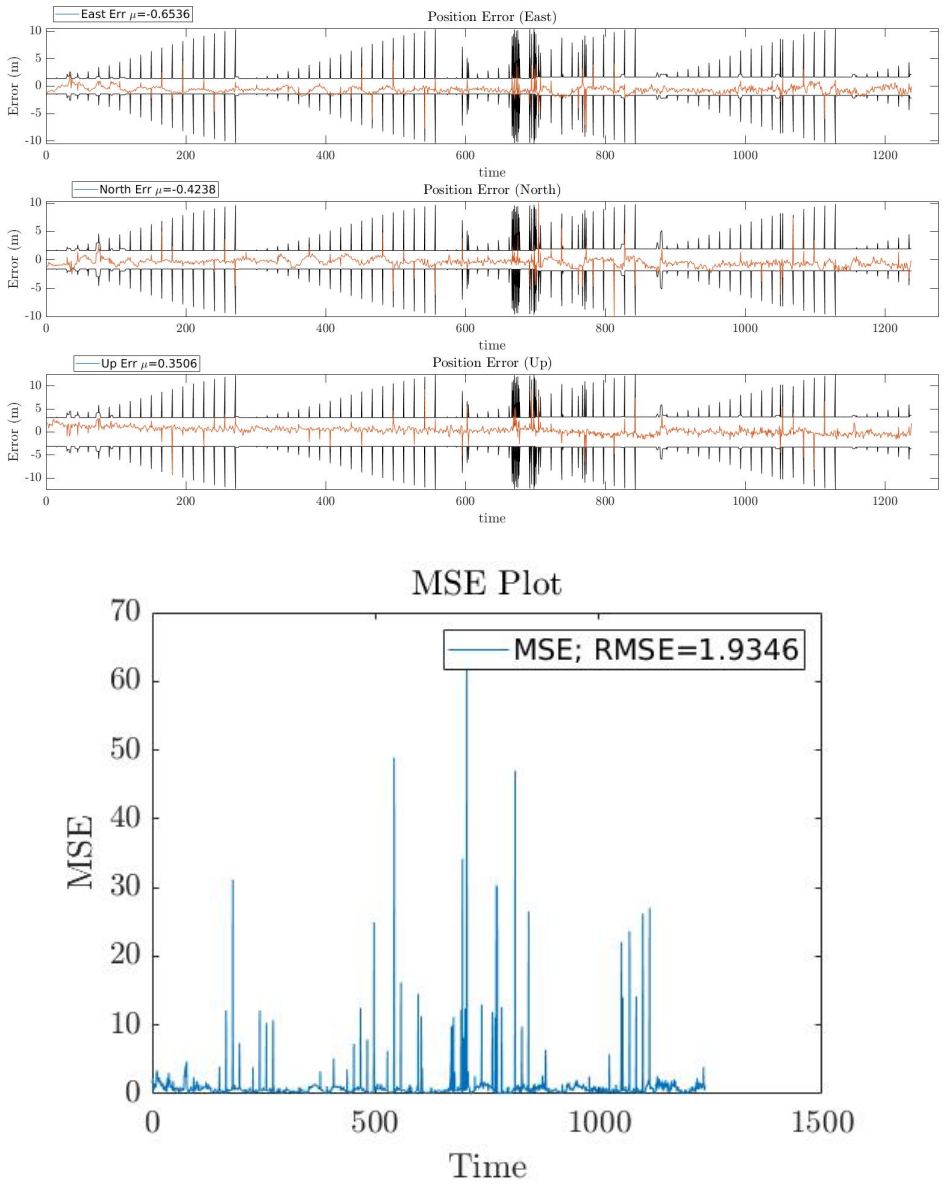
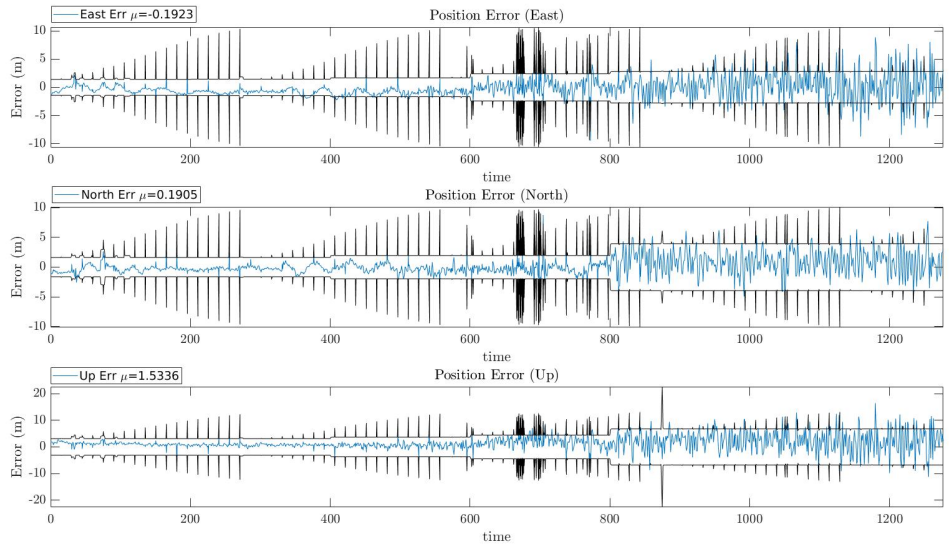


Figure 34: ARMAS's 3D Error and MSE Plot for Experiment 2 (Simultaneous Jamming)

Satellite 4 and 26 drop out at time-step 700 and remain jammed until the end of the data. The covariance bound remained stable at the jamming implementation timestep, leading to the state error and MSE to also remain stable. Statistically, there was no change with the RMSE value for this experiment compared to the Baseline.

Conclusively, ARMAS was not affected at all by the simultaneous jamming scenario and handled it well.



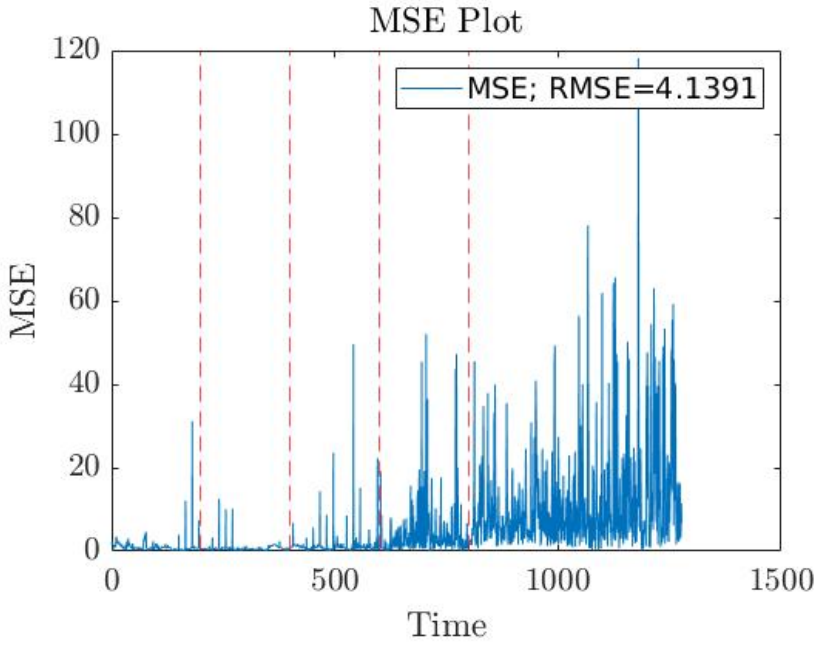
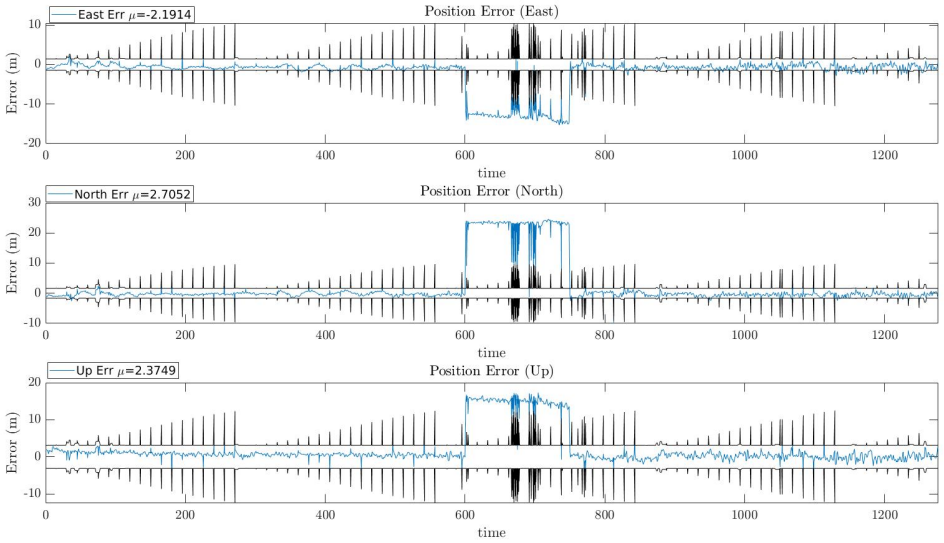


Figure 35: ARMAS's 3D Error and MSE Plot for Experiment 3

For the periodic jamming scenario, Satellite 22, 27, 31, and 9 are affected at different time-steps and remain affected until the end of the data: Satellite 22 affected at time-step 400, Satellite 27 affected at time-step 600, Satellite 31 affected at time-step

800, and Satellite 9 affected at time-step 1000. The covariance bound wasn't affected much by the first two satellites dropping out, but similar to RBRAIM lost most of its confidence in estimating the state after the third satellite was jammed. The state error and MSE were affected when the third satellite was jammed, but even more so when last satellite was jammed resulting in the instability in both plots. ARMAS still handled the scenario well overall with the difference in RMSE value from the Baseline being 2.211 (Table 10), equivalent to RBRAIM's performance. Therefore, ARMAS handled the periodic jamming scenario well overall.



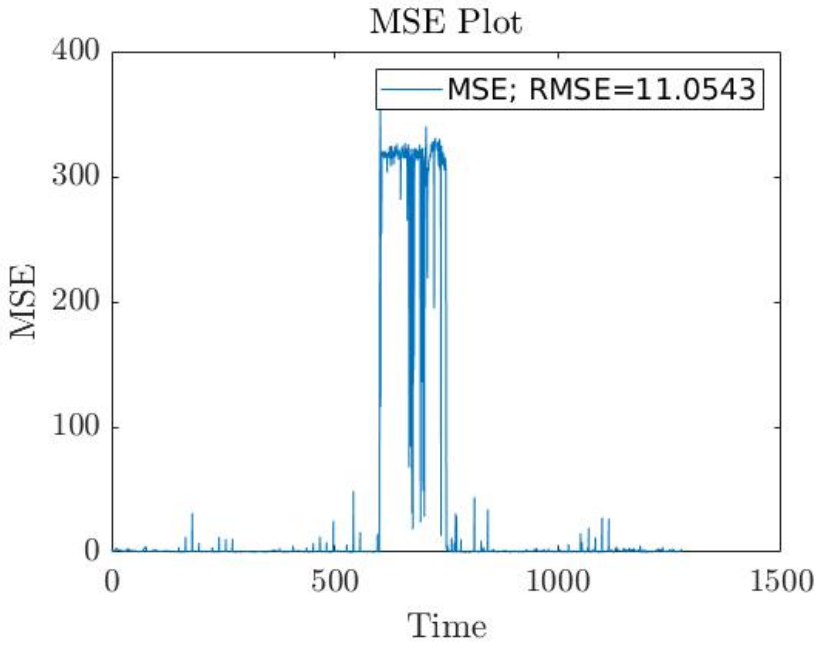
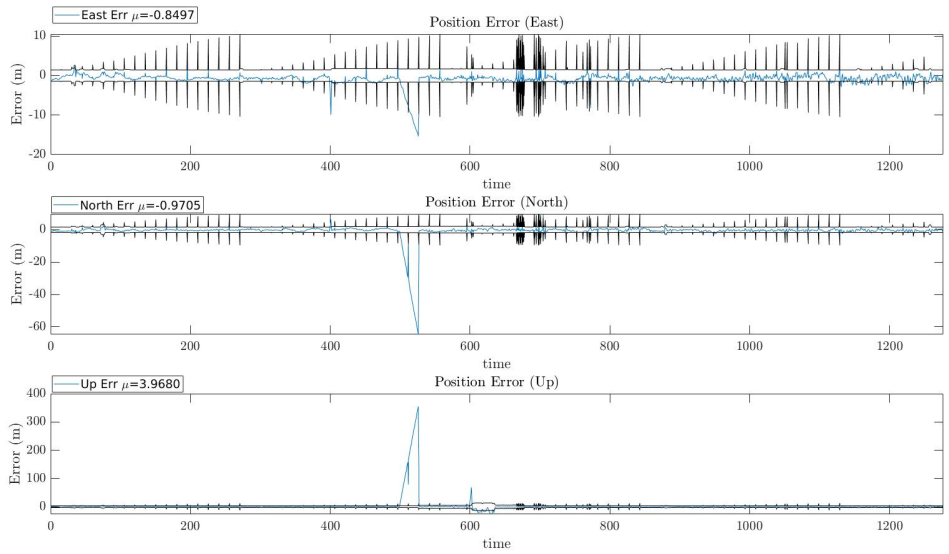


Figure 36: ARMAS's 3D Error and MSE Plot for Experiment 4

A bias of 50 meters is added to Satellite 4 and Satellite 26 at time-step 600. Both of the biases last for 150 seconds before the pseudorange goes back to normal. The covariance bound surprisingly remains stable throughout the entirety of the bias application. However, the state error and MSE plots show a different outcome. The state error for the East, North, and Up coordinate system were hit hard by bias,

signifying the system has trouble handling the bias. The state error also remained high during most of the bias period. There were a few time-steps where the state error drops, showing that ARMAS may have properly removed one or both of the satellites. However, the system was too inconsistent at finding and excluding the affected satellites, resulting in poor state estimations. The MSE plot shows the same outcome as the state error plots. Due to the poor state estimations, the RMSE value for this scenario is well above 8 with a value of 11.054. Therefore, ARMAS did not handle the simultaneous bias scenario well.



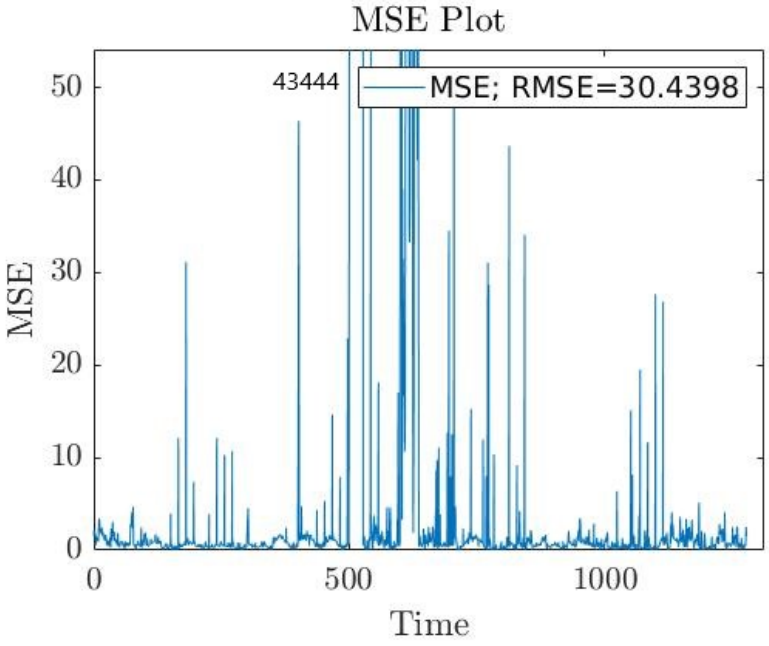


Figure 37: ARMAS's 3D Error and MSE Plot for Experiment 5

A ramp bias of 10 meters is added to Satellite 3, 4, 16, and 26 at varying time-steps: Satellite 3 affected at time-step 300, Satellite 4 affected at time-step 400, Satellite 16 affected at time-step 500, and Satellite 26 affected at time-step 600. The biases last for 100 seconds before the pseudorange returns to normal and the next bias is applied. ARMAS also had a hard time dealing with this scenario. The

covariance bounds were not affected, but the state error and MSE was. The state error spiked a little in each bias implementation. However, the state error spiked significantly at time-step 500 for each coordinate axis, signifying the system had the hardest time detecting and removing that specific bias implementation. Even though the system did terribly producing an accurate state error estimate, the state error eventually returned to normal signifying that the system did remove the biased satellites properly. The MSE, following the state error plots, spiked with each bias implementation, and having the largest spike at time-step 500. This resulted in a huge RMSE value of 30.44, signifying ARMAS struggled during this scenario. Therefore, ARMAS handled the periodic jamming scenario terribly.

Exp.	**Avg x-err**		**Avg y-err**		**Avg z-err**		**RMSE**	
	RB	**AR**	**RB**	**AR**	**RB**	**AR**	**RB**	**AR**
Baseline Sim.	-0.016	N/A	0.114	N/A	0.019	N/A	2.538	N/A
Baseline C.A.	-0.522	-0.629	-0.006	-0.317	0.574	0.456	1.975	1.929
Exp. 1	0.033	N/A	0.022	N/A	-0.118	N/A	2.529	N/A
Exp. 2	-0.3	-0.654	-0.051	-0.424	0.494	0.351	2.09	1.935
Exp. 3	-0.006	-0.192	0.49	0.191	1.287	1.534	4.197	4.139
Exp. 4	-0.271	-2.191	-0.018	2.705	0.549	2.375	1.976	11.054
Exp. 5	-0.331	-0.85	-0.095	-0.971	0.975	3.968	5.222	30.44
Exp.: Experiment RB: RBRAIM AR: ARMAS								

Table 9: Summary of Results for RBRAIM and ARMAS

The table above shows the numerical results produced by both RBRAIM and ARMAS. When the systems performs well in predicting the state of the receiver and is unaffected by the scenario applied, the average error for each axis and the RMSE value barely changes from its Baseline results. However, if it had a relatively difficult time handling the scenario, you will notice a significant change between the experiment results and the Baseline results. The average state error shows which axis the system had a hard time predicting, and the RMSE value quantifies how

well the system over the entirety of the experiment. For RBRAIM, the system had a difficult time handling Experiment 3 (Periodic Jamming) and 5 (Periodic Ramp Bias) compared to the other experiments. However, due to the RMSE value remaining low, it is considered an accurate system. ARMAS struggled with Experiment 4 and 5 with the RMSE value being above 10 for both. Therefore, ARMAS is an accurate system for the jamming scenarios (Experiment 2 & 3), but inaccurate for the bias scenarios (Experiment 4 & 5).

Exp.	**Diff Avg x-err**		**Diff Avg y-err**		**Diff Avg z-err**		**Diff RMSE**	
	RB	**AR**	**RB**	**AR**	**RB**	**AR**	**RB**	**AR**
Exp. 1	-0.049	N/A	0.092	N/A	0.137	N/A	0.009	N/A
Exp. 2	-0.222	0.025	0.045	0.107	0.08	0.106	-0.115	-0.006
Exp. 3	-0.516	-0.437	-0.496	-0.507	-0.713	-1.077	-2.222	-2.211
Exp. 4	-0.251	1.563	0.012	-3.022	0.025	-1.979	-0.001	-9.126
Exp. 5	–0.191	0.221	0.089	0.654	-0.401	-3.512	-3.247	-28.51
Exp.: Experiment Diff: Difference RB: RBRAIM AR: ARMAS								

Table 10: Difference of Baselines for RBRAIM and ARMAS

The table above displays the difference between the Baseline and Experiment values to better compare how accurate the systems were to each other.

Experiment 1, shows that RBRAIM is a functional system, validating that the system is operationally sound. With a subtle difference in the means of the 3D positions and an RMSE of 0.009, the system identified and excluded each bias appropriately without any major impact on the accuracy navigation solution.

In both jamming scenarios (Experiment 2 & 3), the ARMAS system proved to be more accurate than RBRAIM. Overall, both systems handled jammed well. With the simultaneous jamming in Experiment 2, both systems were within 0.3 meters of their average and only off a small change in the RMSE value. In the periodic jamming test (Experiment 3), both systems' RMSE value were statistically equivalent, signifying RBRAIM and ARMAS accuracy performance being the same. The reason

why ARMAS did better overall is due to its design concept: ARMAS's ab initio system is designed to handle asynchronous all-source sensors, not just GNSS measurements. Therefore, during these experiments, the ARMAS system uses position and velocity sensors to aid in the prediction of the UAV state. The more information integrity systems like RBRAIM and ARMAS have available, the more accurate the provided state estimates become.

For both biasing scenarios (Experiment 4 & 5), the RBRAIM system had a significantly higher accuracy overall than ARMAS. In both Experiment 4 and 5, RBRAIM was less the 0.5 meters off from its baseline's average in each axis, while ARMAS provided results had a difference in values up to 3.512 of its baseline's average. Additionally, the difference in RMSE values explicitly shows how much better RBRAIM did than ARMAS. Between the systems RMSE differences, RBRAIM did better by a value of 9.126 in Experiment 4 and 28.51 in Experiment 5. Due to RBRAIM's sequential local test, it has no issue with identifying and excluding biases, even when there are multiple biases being applied in the same time-step. However, the ARMAS system is built upon the assumption of a single fault/satellite failure. This causes degraded performance when biases are being applied to more than one pseudorange measurement. This is attributable to what is euphemistically called "modelling error."

4.3 RBRAIM and ARMAS Functionality

This section discusses the functionality and efficiency of RBRAIM and ARMAS. Specifically, this section will discuss the results of the biasing experiments (Experiment 1, 4, and 5), when the system identified the bias, and if it correctly excluded the biased satellite. Additionally, how big the bias got in Experiment 5 will be discussed. The results shown in this section will also assist in explaining the results in **??** by

pinpointing specifically why there were accuracy state estimation issues during the bias scenarios. The methods of how each system identifies and excludes biased satellites was discussed in the previous chapters: Section 3.3 for the RBRAIM system and Section 2.7 for the ARMAS system. Experiment 1 and 4 serve to test the efficiency of both systems. This is done by observing how long it takes for both systems to identify there is a bias, remove the biased satellite properly, and how long it takes for the system to bring the satellite back in once the bias ceases – referred to as "recovery time." Experiment 5 is used to determine the detection rate by observing how big the bias gets before the system detects an integrity issue.

Table 11 summarizes the results of the steady bias scenarios, Experiment 1 and 4. This is done by displaying the difference in time of when the bias was implemented and flagged, and the showing the difference in time of when the bias ceased and the state estimates recovered. Additionally, Table 11 includes a determination whether the system removed the biased satellite properly. Table 12 shows results similar to Table 11, but also includes a comment on how big the bias got when flagged.

Exp.	**Diff Added & Caught**		**Diff Ceased & Recovered**		**Properly Removed? (Yes/No)**	
	RB	**AR**	**RB**	**AR**	**RB**	**AR**
Exp. 1	0, 0	N/A	0, 0	N/A	Y, Y	N/A
Exp. 4	0, 0	1, 1	0, 0	31, 31	Y, Y	N, N
Exp.: Experiment Diff: Difference RB: RBRAIM AR: ARMAS						

Table 11: RBRAIM and ARMAS efficiency results of Experiment 1 and 4

The table above displays the efficiency and effectiveness performance of RBRAIM and ARMAS for Experiment 1 (RBRAIM Validation) and 4 (Simultaneous Bias).

The results of Experiment 1 validate that the RBRAIM system functions properly and efficiently. RBRAIM caught and recovered from the bias as soon as it was implemented. The system also correctly removed the satellite causing the integrity

issue.

With simultaneous bias (Experiment 4), RBRAIM had the exact same results as Experiment 1. The system had no issue handling the bias efficiently due to its sequential local test. However, ARMAS did not handle simultaneous biasing well. ARMAS did flag the bias very quickly with only 1 time step passing before recognizing there was an integrity issue. Even though the system responded quickly, it did not recover until 31 seconds after the bias on both satellites ceased. This is due to the system not having a way to properly handle more than one at a time. Consequently, ARMAS could not properly identify which satellite being biased leading to the system not being able to properly remove it from the navigation solution. This is what ultimately caused the poor performance in accuracy for ARMAS during Experiment 4 in **??**.

Exp. & Bias	**Diff Added & Caught**		**Bias Value when Caught**		**Diff Ceased & Recovered**		**Properly Removed? (Yes/No)**	
	RB	**AR**	**RB**	**AR**	**RB**	**AR**	**RB**	**AR**
Exp. 5 Bias 1	6	2	60	20	0	96	Y	Y
Exp. 5 Bias 2	6	27	60	270	0	135	Y	Y
Exp. 5 Bias 3	8	27	80	270	0	66	Y	Y
Exp. 5 Bias 4	5	3	50	30	0	59	Y	Y
Exp.: Experiment Diff: Difference RB: RBRAIM AR: ARMAS								

Table 12: RBRAIM and ARMAS efficiency results of Experiment 5

The table above displays the efficiency and effectiveness performance of RBRAIM and ARMAS for Experiment 5 (Periodic Ramp Bias).

For both RBRAIM and ARMAS, the periodic ramp bias scenario was difficult to handle. Both systems properly identified and removed the biased satellite throughout

each timestep. RBRAIM did overall better when it came to identifying the bias within a short time with an average time of 6.5 time steps throughout the experiment compared to ARMAS's 14.75. However, ARMAS beat RBRAIM for 2 out of the 4 biases applied by catching bias 1 and 4 by at least 2 timesteps faster – meaning ARMAS discovered the integrity issue before the bias got higher than 30 meters compared to RBRAIM's 60 meters. For the other 2 biases, ARMAS did significantly worse, with the bias getting up to 270 meters before being identified while RBRAIM caught the biases at 60 and 80 meters. Comparable to Experiment 4, ARMAS still took a lot longer to recover from the bias than RBRAIM, with values as 135 time-steps to recover while RBRAIM put the unbiased satellite back in immediately.

4.4 Summary

This section summarizes the results of the experiments ran and provides an overall analysis of which system was more accurate and efficient during the scenarios. When considering the accuracy and efficiency of the RBRAIM system, it performed much better than ARMAS during bias scenarios. RBRAIM accurately predicted the position state component and efficiently functioned during periodic steady bias, simultaneous steady bias, and periodic ramp bias scenarios. With the prediction of the position being no greater than 0.5 meters off the average and the RMSE being significantly lower than ARMAS's, RBRAIM beats ARMAS when it comes to biasing scenarios. Due to RBRAIM's sequential local test, it was able to properly identify if there was bias in the navigation solution, remove it from the navigation solution, recover from the bias immediately, and continue running effectively unlike ARMAS. However, ARMAS is the system best suited for jamming scenarios compared to RBRAIM. ARMAS technically did better overall during the jamming scenarios, but the difference between both systems were minimal. During the periodic jamming

scenario, RBRAIM and ARMAS systems' performance was identical. With a difference of 0.109 in RMSE value between both the systems, statistically the systems performed equivalent during the simultaneous jamming scenario as well. The reason for ARMAS's slight out-performance during the jamming scenarios is attributable to the system having position and velocity sensors aiding in predicting the states. This allows the ARMAS system to pull in more information.

V. Conclusions

5.1 Overview

This chapter summarizes the research and results obtained during the experiments performed in this study. Section 5.2 restates the significant conclusions found during the experimentation and quantitative analysis. Section 5.3 discusses the contributions of this work to the field of PNT. Lastly, Section 5.4 provides recommendations for future research and improvements of RBRAIM and ARMAS navigation systems.

5.2 Conclusions

By comparing RBRAIM to ARMAS, an analysis of the accuracy, reliability, and effectiveness of the ARMAS framework was successfully accomplished. Conclusively, all five set out objectives of this research were achieved: First, design a high performance and reliable RAIM system; Second, perform a controlled experiment to validate the performance of the new RAIM system developed by using simulated data; Third, examine the reliability and effectiveness of identifying the biased satellite in the RAIM and ARMAS system; Fourth, measure the accuracy of the receiver's position state component provided by both systems during realistic navigation scenarios; Fifth, evaluate and compare the results provided by both systems.

During jamming scenarios (Experiment 2 & 3), ARMAS outperformed RBRAIM by an RMSE value of 0.11, being a more accurate system overall (Table 10). Even though both systems had low errors when predicting 3D position of the receiver, ARMAS was more resilient to this interference. Due to its structure and usage of position and velocity sensors, ARMAS was able to maintain a lower RMSE in both simultaneous jamming and periodic jamming.

For biasing scenarios (Experiment 4 & 5), RBRAIM completely outperformed

ARMAS resulting in a total difference of 34.39 in RMSE value (Table 10), being able to effectively detect and efficiently exclude the bias. RBRAIM's local test handled the simultaneous bias scenario well, consistently maintaining a low bias-flagging time, correctly identifying the biased satellite, and recovering from the bias immediately. However, ARMAS's structure only being able to handle single faults, led to poor performance during the simultaneous bias scenario. During the periodic ramp bias scenario, RBRAIM did not perform as well as it did during the simultaneous bias scenario. But it still detected the bias relatively quick, not allowing it to get higher than 80 meters and removing it properly (Table 12). RBRAIM, similar to the simultaneous bias scenario, recovered as soon as the bias ceased. ARMAS started off well catching the bias when it was only 20 meters. However, it quickly fell off, allowing the next two biases implemented to reach 270 meters before flagging them. Additionally, ARMAS took a while to recover from the bias with an average time of 89 time steps after the bias stopped. Regardless of ARMAS's poor performance in detection rate and recovery, it still correctly excluded all biased pseudorange measurements.

5.3 Research Contributions

This thesis compared the accuracy, reliability, and effectiveness of ARMAS compared to another validated integrity monitoring method. The evaluation of the alternative resilient navigation methods is performed in a quantitative way using well designed experiments and real data.

The performance comparison also yields improvements to the ARMAS framework by showing the weak points of the system and methods to improve it. Lastly, this research provides a resilient RAIM system simulated in Python. The system is to be used for resilient navigation performance comparisons in realistic navigation scenarios, to further improve the ARMAS framework.

5.4 Recommendations for Future Work

This research provides avenues to extending the ARMAS framework and improving it in the pursuit of increased reliability of navigation systems. The following suggestions of future work are based on the insights gained during the performance of this research:

1. Perform an additional comparison using the RBRAIM system built but integrate position and velocity sensors into the structure. This would make for a "fairer" comparison between systems during the jamming scenarios.

2. Design other Inertial Measurement Unit (IMU) aided RAIM methodologies to compare ARMAS. RBRAIM has revealed a few of ARMAS's weak points, but there could be other issues that RBRAIM did not bring up.

3. Implementing ARMAS into Python or RBRAIM into Matlab. It would make for an easier and better comparison when it comes to plotting and quantitative data.

4. Implementing a local test into the ARMAS system. Currently, ARMAS and RBRAIM have a similar global test. If a local test is implemented into the ARMAS framework, it could lead to an improvement in performance.

5. Use the p-value method for global and local testing instead of the chi-square method.

Bibliography

1. United State Government Accountability Office. United states government accountability office defense navigation capabilities dod is developing positioning, navigation, and timing technologies to complement gps. *Report to the Committee on Armed Services, U.S. Senate*, 2021.

2. Heidi Kuusniemi and Gérard Lachapelle. Gnss signal reliability testing in urban and indoor environments. *Proceedings of the National Technical Meeting, Institute of Navigation*, 2004, 2004.

3. Jiaxing Liu, Jiaxing Liu, Mingquan Lu, Zhenming Feng, and Jinling Wang. Gps raim: Statistics based improvement on the calculation of threshold and horizontal protection radius. School of Surveying the University of New South Wales, 2005.

4. D H Titterton and J L Weston. *Strapdown Inertial Navigation Technology, Second Edition*. Institution of Engineering and Technology, 2004.

5. Jonathon S Gipson. *Air-to-Air Missile Enhanced Scoring with Kalman Smoothing*. Thesis, Air Force Institute of Technology, 2012.

6. Elliott D. Kaplan and Christopher J. Hegarty. *Understanding GPS/GNSS Principles and Applications, Third Edition*. Artech House, 2017.

7. ARINC Engineering Services. Navstar gps space segment/navigation user interfaces. Online report for space and missile systems center (smc), 2004.

8. Geoffrey Blewitt. Basics of the gps technique: Observation equations §. University of Newcastle, 1997.

9. M. Mosavi, S. Azarshahi, I. Emamgholipour, and A. A. Abedi. Least squares techniques for gps receivers positioning filter using pseudo-range and carrier phase

measurements. *Iranian Journal of Electrical and Electronic Engineering*, pages 18–26, 2014.

10. Aboelmagd Noureldin, Tashfeen B. Karamat, and Jacques Georgy. *Fundamentals of inertial navigation, satellite-based positioning and their integration.* Springer Berlin Heidelberg, 1 2013.

11. Adriano Solimeno. *Low-Cost INS/GPS Data Fusion with Extended Kalman Filter for Airborne Applications.* Thesis, Instituto Superior Técnico, 2007.

12. Elena Punskaya and St. John's College. *Sequential Monte Carlo Methods for Digital Communications.* PhD dissertation, University of Cambridge, 2003.

13. Ján Mochnáč, Stanislav Marchevský, and Pavol Kocan. Bayesian filtering techniques: Kalman and extended kalman filter basics. *Proceedings of 19th International Conference Radioelektronika*, pages 119–122, 2009.

14. Felix Govaers. *Introduction and Implementations of the Kalman Filter.* Intechopen, 2019.

15. Kamran Iqbal. *Introduction to Control Systems.* University of Arkansas at Little Rock, 2021.

16. Roger R Labbe Jr. *Kalman and Bayesian Filters in Python.* 2015.

17. PB Ober. Integrity prediction and monitoring of navigation systems. *Uniwersytet ślaski*, pages 343–354, 2003.

18. Alban Rakipi, Bexhet Kamo, Shkelzen Cakaj, Vladi Kolici, Algenti Lala, and Ilir Shinko. Integrity monitoring in navigation systems: Fault detection and exclusion raim algorithm implementation. *Journal of Computer and Communications*, pages 25–33, 2015.

19. R. Brown and Gerald Chin. Gps raim: Calculation of threshold and protection radius using chi-square methods-a geometric approach. *Global Positioning System: Inst. Navigation, vol. V*, pages 155–179, 1997.

20. Yuanxi Yang and Junyi Xu. Gnss receiver autonomous integrity monitoring (raim) algorithm based on robust estimation. *Geodesy and Geodynamics*, 7:117–123, 3 2016.

21. Fanchen Meng, Shan Wang, and Bocheng Zhu. Gnss reliability and positioning accuracy enhancement based on fast satellite selection algorithm and raim in multiconstellation. *IEEE Aerospace and Electronic Systems Magazine*, pages 14–27, 2015.

22. J. Cosmen-Schortmann, M. Azaola-Sáenz, M. A. Martínez-Olagüe, and M. Toledo-López. Integrity in urban and road environments and its use in liability critical applications. *IEEE PLANS, Position Location and Navigation Symposium*, pages 972–983, 2008.

23. Nagendra R. Velaga, Mohammed A. Quddus, Abigail L. Bristow, and Yuheng Zheng. Map-aided integrity monitoring of a land vehicle navigation system. *IEEE Transactions on Intelligent Transportation Systems*, 13, 2012.

24. Tareq Binjammaz, Ali Al-Bayatti, and Ashwaq Al-Hargan. Gps integrity monitoring for an intelligent transport system. *10th Workshop on Positioning, Navigation and Communication*, 2013.

25. Hira Imtiaz, Muhammad Haroon Yousaf, Hassan Sajjad Malik, and Muhammad Ali Akhtar. Design and implementation of receiver autonomous integrity monitoring algorithm on dsp for small uav applications. *2019 International Conference on Frontiers of Information Technology*, pages 278–283, 2019.

26. Juan Blanch and Todd Walter. Satellite navigation for aviation in 2025. *Proceedings of the IEEE*, pages 1821–1830, 2012.

27. Mattia Berardo and Letizia Lo Presti. On the use of a signal quality index applying at tracking stage level to assist the raim system of a gnss receiver. *Sensors (Switzerland)*, 2016.

28. Majumdar Panagiotakopoulos. Extreme value theory-based integrity monitoring of global navigation satellite systems. *GPS Solutions*, pages 133–145, 2014.

29. Carlos Daniel Salos Andrés. *Integrity monitoring applied to the reception of GNSS signals in urban environments*. PhD dissertation, Institut National Polytechnique de Toulouse, 2012.

30. Juan D Jurado. *Autonomous and Resilient Management of All-Source Sensors for Autonomous and Resilient Management of All-Source Sensors for Navigation Assurance Navigation Assurance*. PhD dissertation, Air Force Institute of Technology, 2018.

31. Jonathon Gipson. *Collaborative All-source Navigation with Integrity*. PhD dissertation, Air Force Institute of Technology, 2021.

32. Kyle Kauffman, Daniel Marietta, John Raquet, Daniel Carson, Robert C. Leishman, Aaron Canciani, Adam Schofield, and Michael Caporellie. Scorpion: A modular sensor fusion approach for complementary navigation sensors. *IEEE/ION Position, Location and Navigation Symposium*, pages 156–167, 2020.

33. Kenneth A. Fisher John F. Raquet. Precision position, navigation, and timing without the global positioning system. *Air and Space Power Journal*, 25, 2011.

34. Wenbo Wang and Ying Xu. A modified residual-based raim algorithm for multiple outliers based on a robust mm estimation. *Sensors (Switzerland)*, 20, 2020.

35. Jiang Liu, Baigen Cai, Debiao Lu, and Jian Wang. An enhanced raim method for satellite-based positioning using track constraint. *IEEE Access*, pages 54390–54409, 2019.

36. Shanlong Kuang. Geodetic network analysis and optimal design: concepts and applications. *Geodetic network analysis and optimal design: concepts and applications*, 1996.

37. W. Baarda. A testing procedure for use in geodetic networks. *Publications on Geodesy*, page 97, 1968.

38. Mark Wickert and Chiranth Siddappa. Exploring the extended kalman filter for gps positioning using simulated user and satellite track data. *Proceedings of the 17th Python in Science Conference*, 2018.

39. Robert Grover Brown and Patrick Y. C. Hwang. Mathematical description of random signals. *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises*, pages 57–104, 2012.

40. Gene H Golub and Charles F Van Loan. Matrix computations. *Linear Algebra and its Applications*, page 728, 1996.

41. Paul Zabalegui, Gorka De Miguel, Alejandro Perez, Jaizki Mendizabal, Jon Goya, and Inigo Adin. A review of the evolution of the integrity methods applied in gnss. *IEEE Access*, 2020.

42. D. M. Hawkins. Identification of outliers. *Biometrical Journal*, 1987.

43. Mark G. Petovello. *Real-time integration of a tactical-grade IMU and GPS for high-accuracy positioning and navigation.* PhD dissertation, University of Calgary, 2003.

44. R.J. Kelly. The linear model, rnp, and the near optimum fault detection and exclusion algorithm. *NAVIGATION, Volume V, Institute of Navigation*, pages 227–259, 1998.

45. Samuel Ryan. *Augmentation of DGPS for marine navigation.* PhD thesis, University of Calgary.

46. W. F. Caspary, J. M. Rueger, and University of New South Wales. School of Surveying. *Concepts of network and deformation analysis.* School of Surveying the University of New South Wales, 1988.

47. Steffen Schön and Hansjörg Kutterer. Optimal design of geodetic monitoring networks by means of interval mathematics. *Proceedings of the 10th FIG International Symposium on Deformation Analysis*, 2001.

Acronyms

AFIT Air Force Institute of Technology. iv

ANT Autonomy and Navigation Technology. iv, 21

ARMAS Autonomous and Resilient Management of All-source Sensors. iv

DoD Department of Defense. 1

ECA Error Characterization Approach. 20

EKF Extended Kalman Filter. 17

ENU East-North-Up. 6

FD Fault Detection. 19

FDE Fault Detection and Exclusion. iv, 19

GAO United States Government Accountability Office. 1

GDOP Geometric Dilution of Precision. 19

GNSS Global Navigation Satellite Systems. iv

GPS Global Position System. vii, 1, 5

HOW handover word. 9

HPL Horizontal Protection Level. 19

ILS Iterative Least Squares. 12

IMU Inertial Measurement Unit. 80

INS Inertial Navigation System. 5, 14

IODC issue of data clock. 9

IODE issue of data ephemeris. 9

KF Kalman Filter. 16

LR Linear Regression. 13

LSE Least Square Estimation. 5

LTP Local Tangent Plane. 6

MC Monte Carlo. 53

MRA Measurement Rejection Approach. 20

MSE Mean Square Error. 46

NAVWAR Navigation Warfare. 21

NED North-East-Down. 6

NLOS Non-Line of Sight. 20

NMCT navigation message correction table. 9

PDF Probability Density Function. 16

PL Protection Level. 19

PNT Position, Navigation, and Timing. 1, 24

RAIM Receiver Autonomous Integrity Monitoring. vii, 1, 19

RBRAIM Residual-Based Receiver Autonomous Integrity Monitoring. iv, vii, 27

RMSE Root Mean Square Error. iv, 46

RSR Resilient Sensor Recovery. 23

SV space vehicle. 2

TLM telemetry. 9

TOW time-of-week. 9

UAV Unmanned Aerial Vehicle. vii, 2, 25

UTC Universal Time Coordinated. 9

VPL Vertical Protection Level. 20

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
24–03–2022	Master's Thesis	Sept 2020 — March 2022

4. TITLE AND SUBTITLE

Autonomous and Resilient Management of All-Source Sensors for Navigation Integrity: A Comparison and Analysis

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Tate, Niles A, 1st Lt, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

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

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

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-MS-22-M-065

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

Air Force Research Lab Space Vehicles Directorate
Dr. Khanh Pham
3550 Aberdeen Ave. SE
Kirtland AFB, NM 87117-5776
khanh.pham.1@spaceforce.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RVBYC

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

12. DISTRIBUTION / AVAILABILITY STATEMENT

Distribution Statement A. Approved for Public Release; Distribution Unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

When navigating using Global Navigation Satellite Systems (GNSS), multiple/redundant, synchronous pseudorange measurements are readily available. However, when navigating in a GNSS degraded and/or denied region, this is not guaranteed. In response to this challenge, the ANT Center developed a framework known as Autonomous and Resilient Management of All-source Sensors (ARMAS). The ARMAS framework is designed to be resilient towards data corruption caused from mismodeled, uncalibrated, and faulty sensors. This thesis further expands on this work by performing a comparison against a Residual-Based Receiver Autonomous Integrity Monitoring (RBRAIM) scheme using simulated and real flight data to evaluate each systems' performance.

15. SUBJECT TERMS
global navigation satellite systems (GNSS), autonomous and resilient management of all-source sensors (ARMAS), global position system (GPS), position navigation and timing (PNT), residual-based receiver autonomous integrity monitoring (RBRAIM), integrity monitoring, fault detection and exclusion (FDE)

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Robert C. Leishman, AFIT/ENG
U	U	U	UU	102	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636 x4755; robert.leishman@afit.edu