3-2022

Evaluation of the ARMAS-SOM Framework with Real Data

Brandon M. Blakely

EVALUATION OF THE ARMAS-SOM FRAMEWORK WITH REAL DATA

THESIS

Brandon M Blakely, B.S.E.E.

AFIT-ENG-MS-22-M-009

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-MS-22-M-009

EVALUATION OF THE ARMAS-SOM FRAMEWORK WITH REAL DATA

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

Brandon M Blakely, B.S.E.E., B.S.E.E.

March 24, 2022

AFIT-ENG-MS-22-M-009

EVALUATION OF THE ARMAS-SOM FRAMEWORK WITH REAL DATA

THESIS

Brandon M Blakely, B.S.E.E., B.S.E.E.

Committee Membership:

Robert C. Leishman, Ph.D
Chair

Peter J. Collins, Ph.D
Member

Clark N. Taylor,, Ph.D
Member

Abstract

The need for a navigation framework able to fuse all-source sensor information and effectively detect faults and exclude erroneous members has increased dramatically with the expanded use of all-source sensors. Autonomous and Resilient Management of All-Source Sensors (ARMAS)-Stable Observability Monitoring (SOM) is a novel framework which seeks to overcome the weaknesses of previous attempts by combining detection, identification, calibration, model selection, and independent evaluation into a single system [1]. ARMAS-SOM is able to detect faults in all-source navigation systems containing multiple types of navigation sensors (e.g. Global Navigation Satellite System (GNSS), velocity, position, and ranging sensors), while actively observing the stability of ARMAS indicating whether or not its Fault Detection and Exclusion (FDE) capabilities can be trusted. This observability function, known as SOM, enables ARMAS to determine when additional sensor information is required to maintain resiliency. While previously tested with simulated sensor data [2], SOM has yet to be tested with real all-source sensor data. To date, ARMAS has only been tested with real GNSS data [3].

This work expands on previous work [3] by introducing more extensive real data into ARMAS, and tests the capabilities of SOM with real Global Positioning System (GPS), ranging radio, and vision-based position sensor data. This use of real data introduces asynchronous measurements and data dropouts which were not present in previous tests. Through the introduction of additional real data and testing ARMAS-SOM with various test scenarios containing L1 jamming and spoofing, the strengths and weaknesses of the ARMAS-SOM framework are analyzed. Additionally, SOM is employed to determine when to add collaborative ranging radio information to

augment the on-board sensor data and thereby maintain resiliency.

Table of Contents

Page

List of Figures

List of Tables

EVALUATION OF THE ARMAS-SOM FRAMEWORK WITH REAL DATA

I. Introduction

1.1 Problem Background

Due to the increasing possibility of Global Navigation Satellite System (GNSS) access being denied, there has been increased interest in maintaining sensor integrity in the presence of multiple all-source sensors. There have been some attempts in the past (see section 2.4 for a more detailed explanation), however these frameworks are limited to a single sensor or sensor pairs and are unable to perform other capabilities such as calibration and remodeling. Autonomous and Resilient Management of All-Source Sensors (ARMAS) is a novel framework developed with the idea of addressing some of these issues [1].

The goal of ARMAS is to provide an autonomous and resilient framework for multi-sensor navigation [1]. Stable Observability Monitoring (SOM), an addition to ARMAS, improves the monitoring capabilities of ARMAS in that it is able to monitor when the filter requires more sensor information in order to effectively perform Fault Detection and Exclusion (FDE) [4]. Understanding both the motivation for and basic functions of the ARMAS-SOM framework are essential to understanding the research contained herein.

Recent research involved utilizing the ARMAS-SOM framework to fuse collaborative sensor information, monitor the integrity of the system, exclude faulty sensors, and maintain an accurate navigation solution within a GNSS-denied environment [2]. The main simulation within that research monitored the performance of a particular

unmanned aerial vehicle (UAV) within a team of five cooperating UAVs. The simulation consisted of four main zones that were traversed by the team of cooperating UAVs: 1) GNSS available, 2) L1 Jamming, 3) L1 Jamming + L2 Spoofing, and 4) GNSS available. During the entire simulation, the integrity of the onboard sensors were monitored, with erroneous sensors automatically identified and excluded from the filter solutions. The approach also monitored the information provided by collaborative team members and the approach detected and excluded an unreliable team member. This simulation showcased the effectiveness of ARMAS-SOM in identifying faults and utilizing ranges between UAVs to augment the navigation solution and maintain integrity during the entirety of the simulation.

In addition to this research [2], there has been some recent work testing the ARMAS framework with real Global Positioning System (GPS) data [3]. Whereas [2] tested multiple sensors, including ranging sensors, [3] examined the use of L1 satellite data from an actual flight test. Therefore, this work [2][3] provide a foundation to build upon in this research. The former [2] providing the baseline for multiple sensors testing the ARMAS-SOM framework, and the latter [3] providing the baseline for analyzing ARMAS with real data.

1.2 Data Collection

The particular data set collected for this thesis involves 21 minutes and 17 seconds of a flight test performed at Camp Atterbury, Indiana. The test setup involves two mobile vehicles: an aircraft and a car, the former being the main focus of this data. Both of the mobile vehicles are equipped with a ranging radio, and there are two stationary ranging radio units placed in the general flight area (refer to Section 3.1 for more details on the experiment setup). Numerous test scenarios were conducted to test ARMAS-SOM with various sensor combinations and jamming/spoofing events.

Throughout the test scenarios, the truth GNSS data recorded by the plane computer is compared to that of the ARMAS filter.

1.3 Research Objectives

The main research objectives of this thesis may be described as follows:

1. Test ARMAS-SOM with more real sensor data from ANT flight tests
2. Compensate for asynchronous and inconsistent data updates
3. Determine strengths and weaknesses of ARMAS-SOM

Previous research [2][3] was leveraged to test more real sensors in the ARMAS framework, with a focus on ranging radio data, thus fulfilling the first objective. This process began by replicating [3] by adding L1 satellite data from a flight test at Camp Atterbury, IN.

After initiating these sensors, additional sensors were added. These included ranging radios, vision-based position estimates, and simulated velocity sensors. This sensor suite mirrors that of previous simulated work [2], with the main differences being that real L1 and ranging radio data was applied to ARMAS while L2 data was not. Another difference is the addition of a position sensor which provides periodic vision-based position estimates, supplemented with simulated position updates. Previous work demonstrated that ARMAS is able to handle different sensor types in simulated data, this thesis expands on this by testing real flight data.

There were two other major assumptions in [2] and [3] on which this research expands. In previous scenarios, it was assumed that (1) all the data was coming in at a constant rate and (2) each sensor had data recorded at every timestamp. The problem with these assumptions are: (a) real data is asynchronous, there is not a constant rate at which data comes in, and (b) sensors will not always have data at

each timestamp. Therefore, this research not only expands on previous research by testing more real sensors in ARMAS, but it also actually tests the ARMAS framework with asynchronous data, thus fulfilling research objective two. Before, it was affirmed that ARMAS could theoretically work with real-life, asynchronous, and sporadic data, but this research demonstrates the effectiveness in actual tests.

The final research objective involves using the data collected in the various test scenarios to determine the strengths and weaknesses of the ARMAS-SOM framework when tested with real data. The eventual goal is to employ ARMAS in a real-life and real-time situation; hence, this thesis tests ARMAS more extensively to see when it does and doesn't break down.

During each test scenario, FDE functionalities are employed to determine when and if a sensor is faulty. In addition to this, SOM functionalities are applied to determine whether or not there is enough sensor information available to reliably perform FDE. This brings us back to an aspect of the first test objective of this research, namely, to only utilize ranging sensors in a situation in which its data is needed to augment the current sensor information. In other words, if L1 sensors are lost due to a fault, outage, or anything else, SOM will monitor the system and determine when additional ranging radio data is necessary, similar to the work in [2].

1.4 Research Limitations

While this research dives further into ARMAS-SOM and brings us one step closer to employing this in a real situation, there are a few underlying assumptions and/or limitations that must be addressed. These include: (1) ARMAS can only detect a single-simultaneous sensor failure, (2) these tests are conducted using post-processed data, and (3) the nature of collaboration differs from that in previous research [2].

One of the main assumptions of ARMAS in its current state is that it is able to

detect a single-simultaneous sensor failure. In other words, if multiple sensors experience a fault simultaneously, confidence in FDE capabilities is lost. This assumption is tested in this work, and must be an understood limitation of this research.

This second limitation comes from testing ARMAS-SOM via post-processed data. These tests utilize much real data and are performed as if the measurements are coming in real-time, however, ARMAS is still not actually employed in a real-time situation in this project. Performing tests this way gives confidence that ARMAS-SOM would perform in the same way in real-time, yet it will take some more work to test it in an actual situation.

The final difference to note between this research and previous research is the nature of collaboration performed. Previous work tested simulated data in a group of UAVs where there was risk of double counting a measurement giving the illusion of better confidence in the navigation solution than truly present. In this research, collaboration is performed in that the main vehicle is supplementing the onboard sensors with ranges from offboard sensors. Nonetheless, there is no risk of double counting data due to the fact that the other mobile vehicle (the truck) is not relying on the aircraft for its position solution.

1.5 Document Overview

The overall format of this document begins with Chapter II: Background. This chapter digs deeper into the background necessary to understand this thesis and explains the context in which this thesis arises. Chapter III explains the way in which the experiment was setup and how the data was collected. This chapter looks in detail at the hardware and software necessary to collect the data and it also explains the post-processing of the data and testing of the data in Matlab. Chapter IV presents and analyzes the results obtained from the various scenarios which were conducted

during the course of this thesis. The final chapter, Chapter V, draws some conclusions in light of these results and touches on some future work necessary to improve on the current research.

II. Background and Literature Review

2.1 Background Overview

The focus of this chapter is on the relevant background information necessary to understand the context in which this research is focused. First, various navigation methods will be touched on, noting some of their strengths and weaknesses, laying out the need for alternative navigation with integrity and reliability. Furthermore, the ranging radios utilized in this research will be introduced, and recent research related to Autonomous and Resilient Management of All-Source Sensors (ARMAS)-Stable Observability Monitoring (SOM) and resilient collaborative all-source navigation will be explained to clarify the exact context of where this research falls. This will be accomplished by explaining the ARMAS-SOM framework by giving a brief description of its development, and introducing in more detail recent publications analyzing the ARMAS framework [3][2]. This thesis builds upon these works by introducing more real data into the equation.

2.2 Common Navigation Methods

Numerous types of positioning systems are routinely employed in the realm of navigation. Some of these positioning systems are intended to obtain absolute position (i.e. Global Navigation Satellite System (GNSS), MagNav [5][6][7], vision-based [8], and others [9]) while others are able to aid in obtaining the relative position (i.e. inertial measurement unit (IMU)). It is important to note that while these GNSS-alternative methods prove useful in specific situations, none of them provide an all-encompassing alternative to GNSS. For this reason, alt-nav systems typically combine data from various sensors (all-source navigation) to obtain a more accurate and reliable position estimate which can be sustained in even the most challenging

situations. The upcoming sub-sections will focus on describing the navigation sensors which are or will likely be employed within ARMAS.

2.2.1 All-Source Navigation

All-source navigation is when various navigation methods (e.g. GNSS, MagNav, visual, and ranging) are fused together in order to obtain a better, more reliable position estimate than would be possible with either of these sensors alone. All-source navigation proves to be particularly useful in the event of GNSS failure or outage for whatever reason. In the case that GNSS information is lost, jammed, or spoofed, a system that relies only on GNSS, or even fewer all-source sensors, will typically have less chance of providing sustainable and accurate position estimate. No sensor alone has proven to be as effective as GNSS in obtaining and sustaining an accurate absolute position solution, which introduces the need for various types of all-source sensors to further aid in a navigation solution.

It is also helpful to have various types of all-source sensors which complement each other. For example, an inertial system is effective in providing a relative position update for a vehicle, however, the longer an inertial system is relied on without reference to an absolute positioning source, the more the error is compounded [10]. In this case, having one or more absolute positioning sensors (i.e. MagNav, imaging, ranging to known location, etc.) in addition to the inertial system will constrain the solution from perpetual drift.

While having additional sensors to aid in a navigation solution is often helpful, it also introduces more opportunities for faulty sensors to negatively affect the results of the navigation solution. This is where the ARMAS-SOM framework comes into play. ARMAS monitors the integrity of each sensor, and will remove faulty sensors which are providing erroneous information. This gives confidence that the navigation

solution provided is not corrupted by faulty information. SOM monitors the system and produces a warning if more sensor information is needed. In this research, as well as with previous research [2], SOM is used to determine when to add collaborative sensor information.

2.3 Ranging Radios

One of the significant contributions of this thesis is the addition of real ranging radio data into ARMAS. This is done to further demonstrate ARMAS-SOM's capabilities in being employed in real-life situations. The ranging radios which were used to gather the data for this thesis are the P440 radios developed by TDSR. This section will briefly introduce the process by which these radios were selected, and how they were configured to collect the ranging data.

2.3.1 Choosing the Ranging Radio

A necessary requirement for this work was to have a network of time-synchronized radios to maximize the efficiency of the network and to ensure the provided timestamps were valid. Although data was collected with two types of ranging radios, this thesis focuses on data obtained by the P440 radios. The reason for this is fourfold: (1) familiarity with the P440s, (2) customizability of P440s made easier via the provided API documentation, (3) consistency of the P440 data, and (4) pricing of the P440s. These four considerations made it fairly easy to choose the P440s for collecting the data to use in ARMAS for the purposes of this thesis.

2.3.2 Two-Way Range Calculation

Before diving further into the specific settings for the P440 in this thesis, it is helpful to understand how the P440 obtains a range calculation. In RangeNet mode

(see Section 2.3.3), each range measurement between any two radios requires one requester radio and one responder radio. The requester initiates the range by sending a packet to the responder radio. The responder responds by transmitting a packet back to the requester radio. When each of these radios transmits a packet, it knows at what time it occurs. Therefore, the range is estimated by taking into account speed of light and the time of flight. This is represented by equation 2.3.1 below.

$$R = \frac{ct}{2} \tag{2.3.1}$$

In this equation, c is the speed of light, R is the Precision Range Measurement (PRM), and t is the time of flight. According to the P440 documentation, this results in sub-centimeter range accuracy [11].

2.3.3 RangeNet Time Division Multiple Access (TDMA) Mode

The P440 radios are capable of operating in three main operating modes: Monostatic Radar Module (MRM), Channel Analysis Tool (CAT), and RangeNet (RN). For this project, only RangeNet mode was utilized for ranging purposes. Within RangeNet mode, there are three submodes: ranging, networking, and localization. Networking mode was selected because it meets the requirements mentioned above; namely, the ability to have multiple time-synchronized ranging radios in a network and greater flexibility in defining a maximally efficient network. This particular networking mode is referred to as TDMA mode. This mode allows the user to define a "slot map" which is a data table which identifies (1) the requester and responder radios ranging with one another in a particular time slot, (2) the type of data which will be sent in this slot (i.e. ranging info or data packet info), and (3) other parameters such as the Pulse Integration Index (PII), antenna mode, and more. The specific slot map definition for this network is described below.

2.3.3.1 Slot Map Definition

There were four P440 radios in the network for which this data was collected. This allows for up to two simultaneous range conversations to occur. For example, radio one and radio two can have a range conversation on one channel while radios three and radio four have a range conversation on a different channel. This increases the efficiency of the system by nearly twofold in comparison to performing one range conversation during each time slot.

In this slot map definition, the radio on the plane is both the master radio (the radio which syncs the clocks of all the other radios) and is the requester radio for each of the ranges. This radio alternates ranging with the three other radios, while also sequencing through each of the 4 antenna polarizations. These are represented by "AA", "AB", "BA", and "BB", where the first letter is the antenna mode of the requester radio and the second letter is the antenna mode for the responder radio. "A" refers to the first antenna port on the P440 while "B" refers to the second port (see Figure 1) [12].

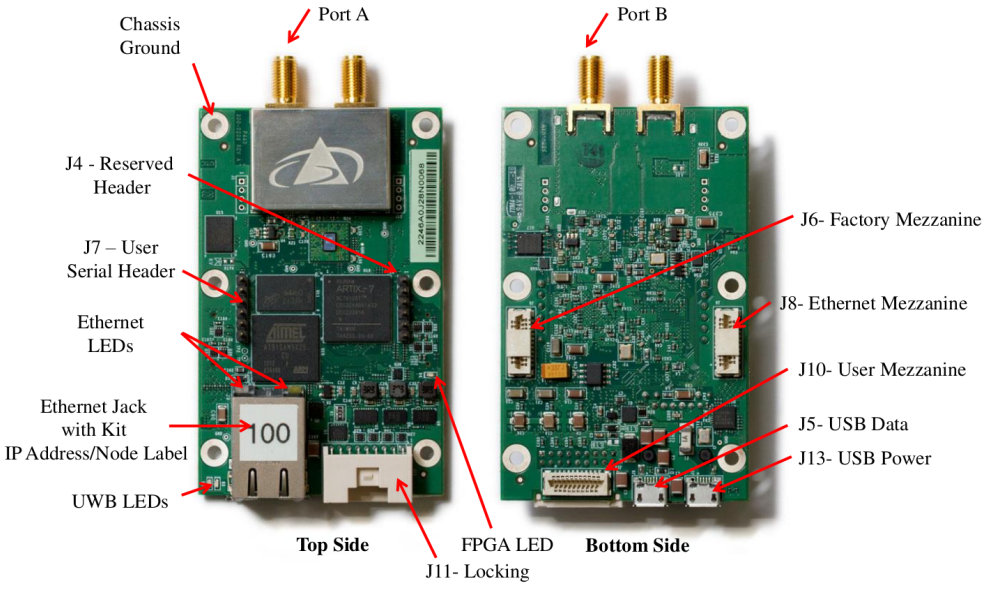

Figure 1: Picture of front and back of a P440 unit, with labels [12]

The advantage of alternating between pols has to do with the antenna beam pattern for these antennas. Each antenna is a dipole and therefore has an antenna pattern represented by a "donut" shape. See Figure 2 for a general 3D antenna beam pattern for a dipole antenna [13], taking note of the nulls at the top and bottom of the antenna. Also see Figures 3 and 4 for the azimuth and elevation beam patterns for the specific dipole antenna used with the P440s [14].

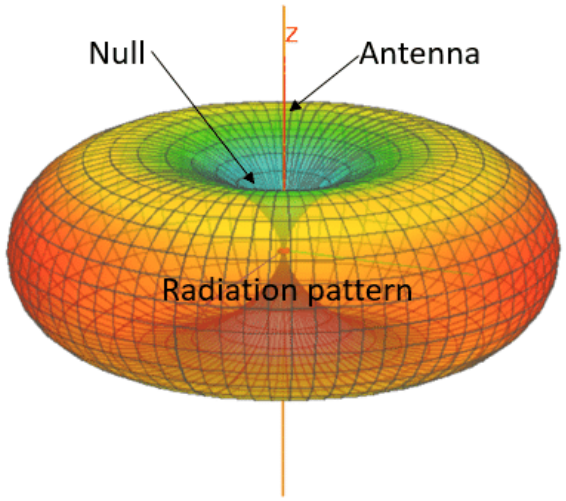

Figure 2: 3D representation of a general dipole antenna pattern [13]

By looking at these plots it is easy to see that while these antennas have great azimuth coverage, the elevation beam pattern is limited, particularly at 0 and 180 degrees. Therefore, having each of the two antennas on the P440s offset by 90 degrees and simultaneously alternating between the 4 polarizations allows for the maximum likelihood of a successful ranging conversation due to staggering these null points in these antennas.

While seeing the antenna patterns gives a general theoretical understanding of why it would be beneficial for the antennas to be placed at different orientations, Figures 5 to 7 demonstrate the practical benefits of utilizing all four polarizations in

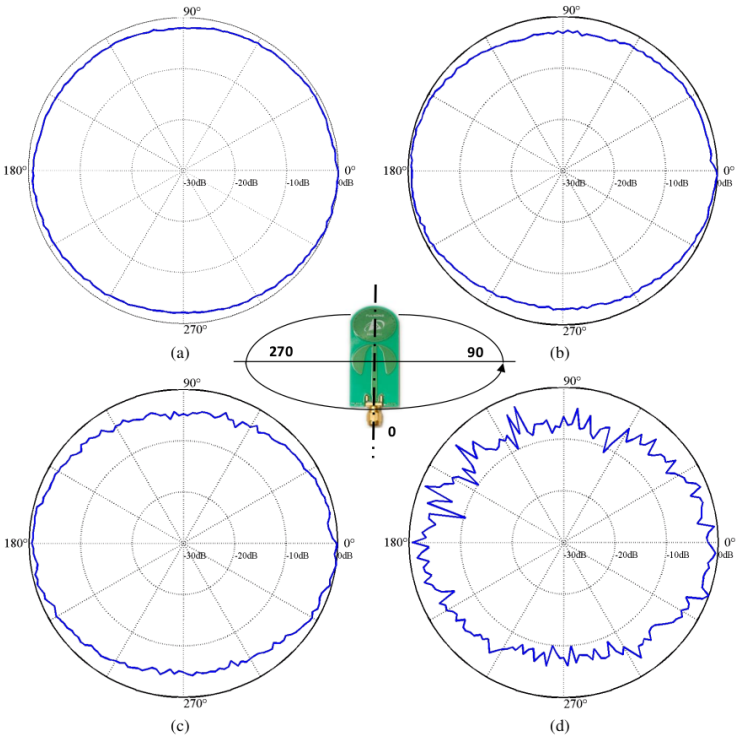

Figure 3: Azimuth beam pattern for the Broadspec dipole antenna at (a) 3 GHz, (b) 4 GHz, (c) 5 GHz, and (d) 6 GHz [14]

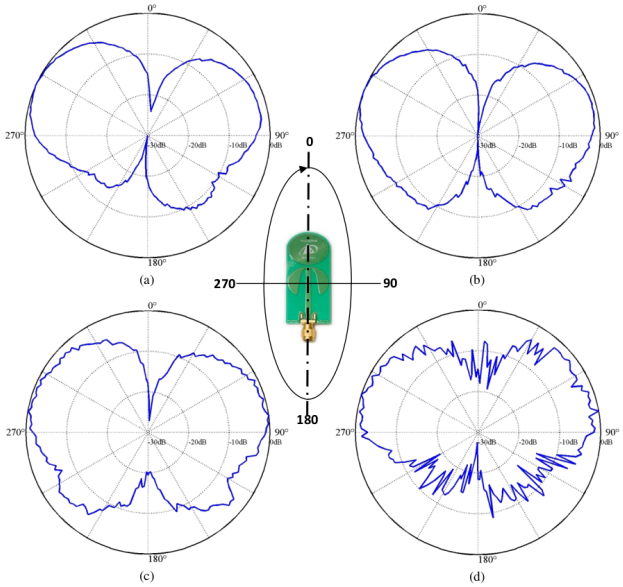

Figure 4: Elevation beam pattern for the Broadspec dipole antenna at (a) 3 GHz, (b) 4 GHz, (c) 5 GHz, and (d) 6 GHz [14]

the application of this thesis. In these Figures, you can see that if all four pols were not used (each pol indicated by a different color dot), or even just two pols were used, the number of successful ranges would be significantly lower. In other words, imagine how the plot would look like if one or more of the colors were removed, there would not be nearly as many data points.

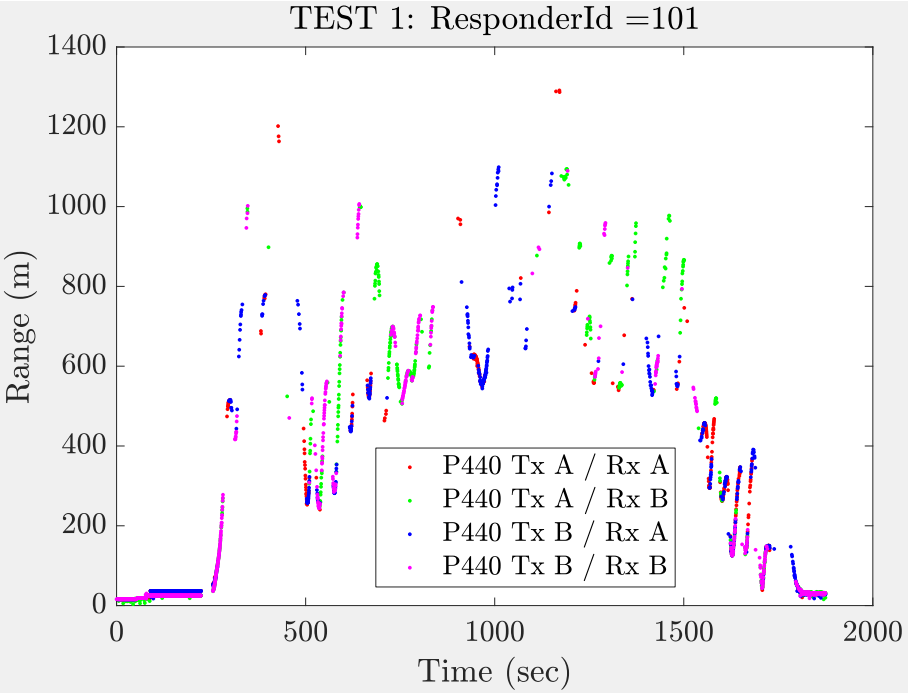

Figure 5: Range plot between master radio and radio 101 with different colors for 4 polarizations

For a more detailed look at the specific slot map definitions for the radios in this thesis, refer to Section 3.1.2 below and the corresponding Figures in this section. For more detailed information on the P440s refer to [12][14][11][15][16].

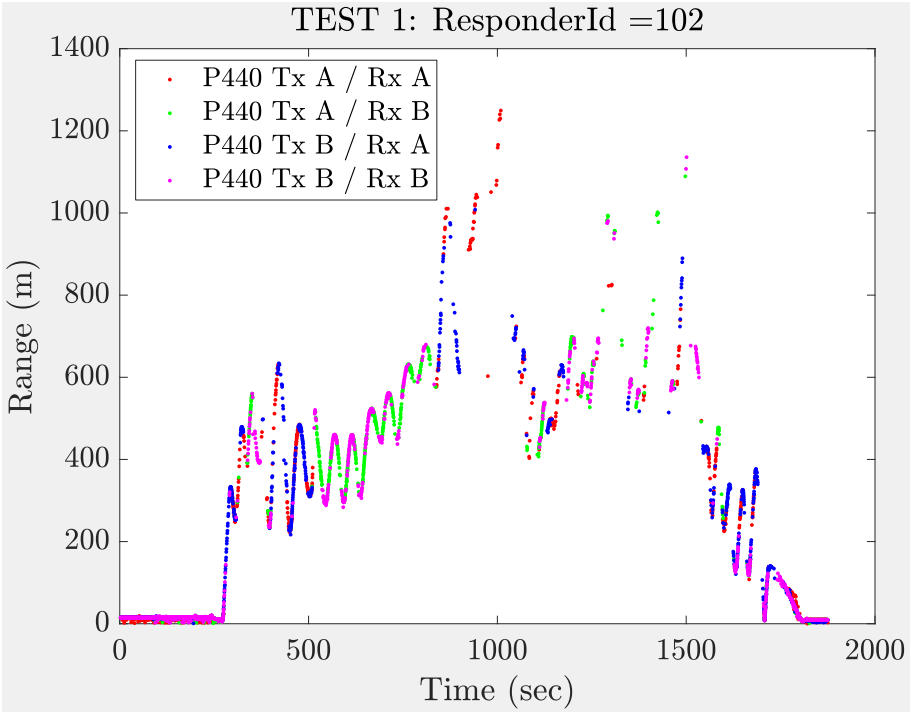

Figure 6: Range plot between master radio and radio 102 with different colors for 4 polarizations

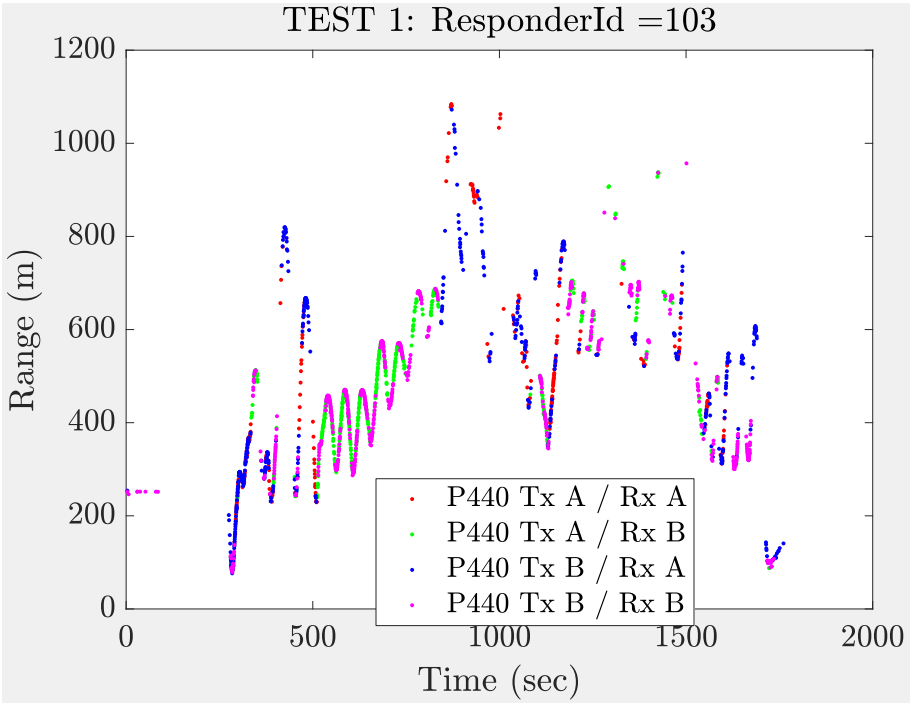

Figure 7: Range plot between master radio and radio 103 with different colors for 4 polarizations

2.4 Relevant Research Background

In the past, there has been some research seeking to address the difficulties of maintaining sensor integrity in the presence of multiple all-source sensors. The main focus of this section is on ARMAS-SOM, which is a more novel attempt at maintaining integrity in a system that uses many types of sensors for navigation. In this section, the general idea of ARMAS and the motivations for ARMAS will be described, followed by following some developments within the ARMAS framework including descriptions of Sensor-Agnostic All-Source Residual Monitoring (SAARM) and SOM which add additional sub-filter levels to further improve the monitoring capabilities of ARMAS.

2.4.1 ARMAS Overview

Much of this overview of the ARMAS framework is drawn from the papers produced by Juan Jurado [1] [17] and Jonathon Gipson [10]. As previously explained, due to the increased interest in alternative navigation systems utilizing multiple sensors, a desire to ensure real-time accuracy and resilience of sensors is highly desired. Previously researched methods have focused on areas such as fault detection and exclusion [18][19][20][21][22][23][24][25][26][27], calibration of sensor models [28][29][30][31][32][33][34][35][36][37][38], and adaptive estimation [39][40][41][42][43][44].

Contemporary research methods with resilient sensor fusion have some limitations. For example, while these fault detection methods are able to detect sensor faults, their applications are limited to a single type of sensor (like GNSS) or a pair, (GNSS and IMU), they do not extend to general sensor types. Furthermore, current online calibration techniques tend to focus on a single sensor and do not usually discuss being able to determine when calibration is needed and the effectiveness of the calibration [1].

Another method which was previously researched is referred to as adaptive estimation. This method seeks to change the sensor model in response to the environment or to take into account missing parameters. The weaknesses related to this method is that it is focused on specific modes of failure and does not perform calibration. In addition to this, it cannot independently determine the validity of the adaptive estimation results [1].

ARMAS was created in an effort to overcome the limitations of these current methods. "[ARMAS] combines detection, identification, calibration, model selection, and independent validation into four interconnected modes of operation: monitoring, validation, calibration, and remodeling." [1]. Each of these 4 modes of operation is briefly described in Table 1.

Mode	**Online Sensor Management Functions**
Monitoring	Detect inconsistent measurement statistics Identify the affected or faulty sensor Decide when to modify existing sensor model
Validation	Validate a questionable sensor model against known sensors Verify results from parameter estimation and model selection Recover from temporary sensor failures
Calibration	Estimate selected sensor model parameters Dynamically augment and initialize necessary filter states Follow a prescribed sequence for parameter estimation
Remodeling	Select best sensor model from a list of candidates Dynamically initialize multiple-model filter bank

Table 1: Description of each of the 4 modes within ARMAS [1]

These four modes are intended to cover the areas of weakness of previous methods. The location of each of these modes in the ARMAS framework is outlined in Figure 8. The overall goal of this framework is to provide an autonomous and resilient framework for multi-sensor navigation which is able to combine these four capabilities into a single framework. For further detail on each of the four operating modes of the ARMAS system, refer to [1].

A state diagram that shows the logical flow of the ARMAS framework is shown in Figure 8. The system is initiated by placing each sensor into an initial operating mode and with an initial designation of "trusted" or "untrusted", based on the reliability of the sensor. If the sensor is "trusted", it enters monitoring mode. However, if it is "untrusted" it enters validation mode. In monitoring mode, ARMAS will monitor the various sensors in an attempt to detect faulty sensors, and it is only in this mode that the sensor will affect the filter solution. If a sensor is detected as faulty, it will immediately exit monitoring mode and be placed into validation mode. If the sensor is only experiencing a temporary fault, it will pass validation once the fault is removed and re-enter monitoring mode. However, if this does not work, ARMAS will look to re-calibrate the sensor. If re-calibration fails, then ARMAS will attempt to remodel the sensor. If remodeling fails, then the sensor is considered failed and will either stay in the failed state or re-enter monitoring mode once the sensor is validated as working properly again.

The overall purpose of ARMAS has remained the same, namely, to provide an autonomous and resilient framework for multi-sensor navigation. One of the most important aspects of this framework includes Fault Detection and Exclusion (FDE) capabilities. During the course of testing ARMAS, various weaknesses were exposed that required some additions to the framework, specifically in the monitoring block of the framework. Two of the main additions include SAARM and SOM. In short, SAARM came first, seeking to improve ARMAS's FDE capabilities for all-source sensors and SOM was developed later in an effort to augment the capabilities of SAARM by adding the capability of detecting when a system needs more sensor information to maintain integrity. Both SAARM and SOM will be explained in more detail in the following subsections, followed later by a brief overview of two important works which are closely related to this thesis.

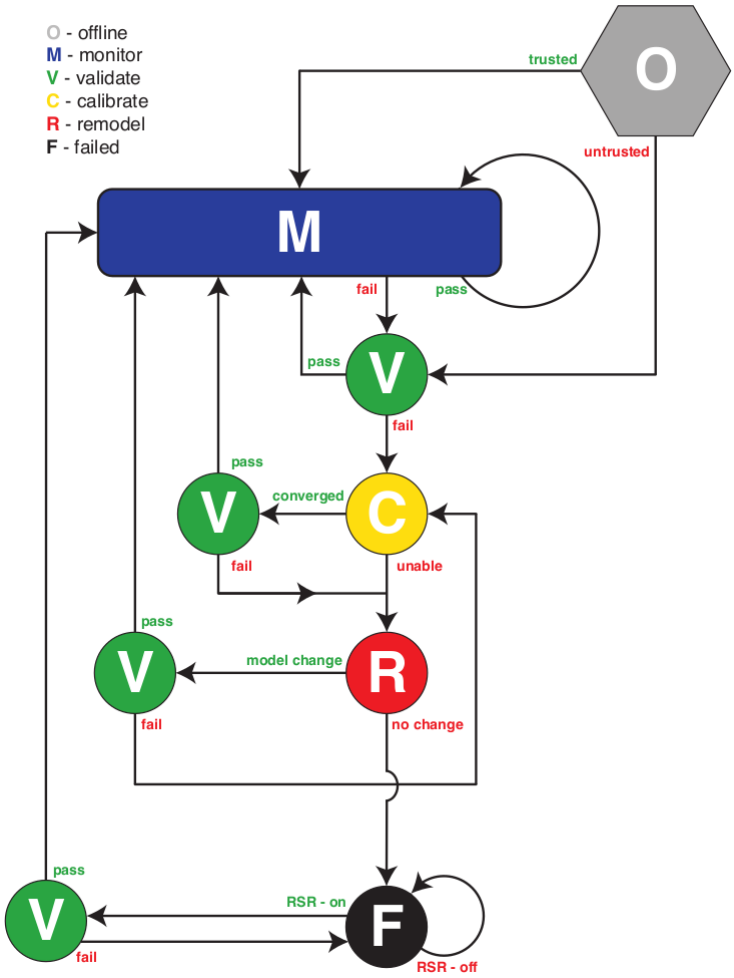

Figure 8: State diagram of the ARMAS framework

2.4.1.1 Mathematical Overview ARMAS

Now that the motivations for ARMAS have been laid out, along with a general overview of what it does, the mathematics behind this framework will be briefly explained. The sources [10][17] provide a more thorough development, this summary is drawn from those works. The overall framework of ARMAS seeks to solve the nonlinear navigation problem

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left[x(t), \epsilon(t), \mathbf{u}(t), t\right] + \mathbf{G}(t)\mathbf{w}(t) \tag{2.4.1}$$

where x is a vector which corresponds to the vehicle's position, velocity, and attitude. ϵ is an $M \times 1$ vector which contains the error states, $\mathbf{u}$ is the control input vector, $\mathbf{G}$ is an $(N+M) \times W$ linear operator, and $\mathbf{w}$ is a $W \times 1$ Gaussian white noise process containing a $W \times W$ continuous process noise strength matrix $\mathbf{Q}$. Measurements from available sensors are updated and the state estimates are propagated utilizing the non-linear state dynamics model. The j^{th} sensor's measurement model is given by equation 2.4.2.

$$\mathbf{z}_k^{[j]} = \mathbf{h}^{[j]}\left[\mathbf{x}(t), \epsilon^{[j]}(t), \mathbf{u}(t), t, \mathbf{p}^{[j]}\right] + \mathbf{v}_k^{[j]} \tag{2.4.2}$$

In this equation, t represents continuous time, while k represents the discrete time index. $\mathbf{h}^{[j]}$ is the nonlinear measurement function for the j^{th} sensor, ϵ^j is an $L \times 1$ subset of ϵ containing the necessary additional states to process sensor j's measurements, $\mathbf{p}^{[j]}$ is a $P \times 1$ vector consisting of model parameters that may be required to process measurement j, for $\mathbf{h}^{[j]}$, and $\mathbf{v}_k^{[j]}$ is a $Z \times 1$ discrete Gaussian white noise process modeled with a covariance of $\mathbf{R}_k^{[j]}$. The j^{th} sensor's $Z \times 1$ residual $\mathbf{r}_\mathbf{j}^{[\mathbf{j}]}$, shown in equation 2.4.3, is calculated given the estimated values $\hat{\mathbf{x}}_\mathbf{k}^-$, $\hat{\epsilon}_\mathbf{k}^{[\mathbf{j}]^-}$, and $\hat{\mathbf{p}}^{[\mathbf{j}]}$.

$$\mathbf{r}_k^{[j]} = \mathbf{z}_k^{[j]} - \mathbf{h}^{[j]}\left[\hat{\mathbf{x}}_k^-, \hat{\epsilon}^{[j]-}, \mathbf{u}_k, t_k, \hat{\mathbf{p}}_k^{[j]}\right] \tag{2.4.3}$$

This residual vector follows a Gaussian distribution,

$$\mathbf{r}_k^{[j]} \sim \mathcal{N}\left(\mathbf{0}_{Nx1}, \mathbf{S}_k^{[j]}\right) \tag{2.4.4}$$

$$\mathbf{S}_k^{[j]} = \mathbf{H}_k^{[j]}\mathbf{P}_k^-\mathbf{H}_k^{[j]^T} + \mathbf{R}_k^{[j]} \tag{2.4.5}$$

where $\mathbf{P}_k^-$ is the $(N+M) \times (N+M)$ state estimate error covariance matrix at time

t_k and $\mathbf{H_k^{[j]^T}}$ is the $Z \times (N + M)$ Jacobian of $\mathbf{h^{[j]}}$

2.4.2 Sensor-Agnostic All-source Residual Monitoring Description

SAARM is an addition to the ARMAS framework which consists of an additional layer of sub-filters [45]. SAARM "makes use of a sum of squared residual covariance Mahalanobis distances as a moving average χ^2 test"[3], which improves on the normalized solution separation method utilized in previous work. The advantage of SAARM is that it is able to work well with sensors of arbitrary types, as long as there is overlapping observability. The ARMAS-SAARM framework has recently been tested with real Global Positioning System (GPS) data, an overview of this test is found in Section 2.6.1. For a more in depth consideration of the SAARM algorithm, refer to [45].

2.4.3 Stable Observability Monitoring Description

In short, SOM adds an observability layer to ARMAS-SAARM. In other words, it augments ARMAS-SAARM by adding an additional layer of filters to help ARMAS know when more sensor information is required to maintain overlapping state observability, which is a key assumption for FDE resiliency. This additional layer operates within the monitoring mode of ARMAS, indicated by the big blue "M" in Figure 8.

Running more tests with ARMAS-SAARM revealed the need for an observability functionality in ARMAS (i.e. SOM). SAARM was initially developed with the assumption that state observability would be simple to maintain with all-source sensors[4]. However, initial testing found this to not be the case [3], as many all-source sensors do not satisfy this assumption of fully overlapping state observability. The other benefit SOM provides is the ability to determine when to add collaborative information to ensure integrity and accuracy [4]. An overview of the simulated tests

performed with the ARMAS-SOM framework is found in Section 2.6.2. For a more in depth consideration of SOM, refer to [4].

2.4.4 Collaborative and Proprioceptive Split-Approach to Navigation

Before diving into the most recent ARMAS tests with simulated data and real flight data, some key concepts must be understood. The first involves the use of both a collaborative and proprioceptive instance of ARMAS.

The collaborative and proprioceptive instances of ARMAS are differentiated by their use of offboard sensors. The proprioceptive solution relies solely on sensors onboard the vehicle while the collaborative solution supplements onboard sensor information with one or more offboard sensors. In this research, anytime a ranging radio sensor is used, collaboration is performed. So the difference between the two solutions is whether or not ranging radios are used to augment onboard sensors.

This split approach which runs a collaborative and proprioceptive instance of ARMAS simultaneously allows for easy comparison to display the effectiveness of collaboration, prevents double-counting from occurring when sharing data with other members, and maintains a fallback option to recover from collaboration [2].

2.5 Measurement Metrics

Three main measurement metrics are deployed to compare the effectiveness of the ARAMS-SOM framework between the collaborative and proprioceptive instances. These metrics include the Root Sum of Squares (RSS), Normalized Estimation Error Squared (NEES), and Guaranteed Position Zone (GPZ). Each of these metrics provide a unique insight into the effectiveness and reliability of the filters, and must be understood to make sense of this research. Brief introductions to each of these measurement metrics are below.

2.5.1 Root Sum of Squares

The purpose of measuring the RSS is to determine how accurate the filter's estimated position is in reference to the truth. The filter's output and the truth data are both expressed in an East-North-Up (ENU) local coordinate frame. The RSS can be calculated simply by calculating the norm of the difference between the estimated and truth ENU coordinates. This is represented by

$$RSS = \sqrt{(E_f - E_t)^2 + (N_f - N_t)^2 + (U_f - U_t)^2}, \tag{2.5.1}$$

where E_f, N_f, and U_f represent the filter's estimated ENU coordinates and E_t, N_t, and U_t represent the truth ENU coordinates. In addition to measuring the RSS, which factors in the ENU coordinates in one measurement, the ENU coordinates are compared individually between the collaborative and proprioceptive instances of ARMAS-SOM.

2.5.2 Normalized Estimation Error Squared

While the RSS measures data corresponding to the first moment of error, the NEES is able to provide information about the second moment of error. This is accomplished by calculating "the square of the distance between the state estimate $\hat{\mathbf{x}}$ and the truth $\mathbf{x}$, normalized by the error covariance $\mathbf{P}$" [10]. This is represented by

$$\gamma_i = (x_i - \hat{x}_i)^T \mathbf{P}_i^{-1} (x_i - \hat{x}_i) . \tag{2.5.2}$$

The NEES metric "provides a quantitative assessment of the estimators credibility from pessimistic to optimistic" [10] [17]. A value lower than 1 corresponds to pessimistic, while a value greater than 1 corresponds to optimistic. Taking the average of the NEES

$$\bar{\gamma}_i = \frac{1}{nm} \sum_{i=1}^{n} \gamma_i \tag{2.5.3}$$

helps determine whether the estimator is credible or not [10] [46]. In this equation, m is the number of dimensions in the state estimate while n is the total number of measurements [10]. In sum, if the average NEES is near 1, then it is likely that the estimator is reliable.

2.5.3 Guaranteed Position Zone

The final measurement metric employed in this thesis is referred to as the Guaranteed Position Zone (GPZ). In essence, the GPZ visualizes "the union of the FDE layer solutions" [10], or in other words, it is a union of all the covariances from the subfilters that provide the FDE capability. Monitoring can help determine whether or not collaboration is improving the size of the guaranteed position zone. Since this is a union of the FDE layer solutions, it is typical for this zone size to decrease as more sensors are added and confidence improves on the location of the vehicle. It is expected for collaboration to provide a notable improvement in the GPZ, as demonstrated in Section 2.6.2. The GPZ is explained more thoroughly in the paper explaining SAARM [45].

2.6 ARMAS Testing Background

Now that a background of the ARMAS-SOM framework and other key concepts related to its use (i.e. collaborative vs. proprioceptive instances, measurement metrics, etc.) have been explained, two specific tests involving the ARMAS framework will be explored. The first, tests ARMAS-SAARM (without SOM) with real GPS flight data. The latter tests ARMAS-SOM with simulated data, using various all-source sensors while performing collaboration.

2.6.1 ARMAS-SAARM Test with Real L1 Satellite Sensors

The initial tests with ARMAS-SAARM were accomplished using simulated sensor data. However, testing ARMAS with actual sensor data did not occur until later [3]. This research which tests ARMAS-SAARM with real GPS data, tests ARMAS-SAARM's capabilities in: (1) successfully fusing sensor data from multiple satellites to get an accurate position and (2) removing a faulty sensor with real-life data. This work successfully demonstrated that ARMAS-SAARM is effective not only with simulated data, but also with real flight data by utilizing the ARMAS framework for navigation and FDE. During this test, a growing bias was added to a single satellite's measurements. The ARMAS-SAARM framework was able to promptly detect this fault and exclude this erroneous sensor. One of the main purposes of this thesis is to further expand on this work by adding more real sensors and developing more scenarios to further demonstrate the effectiveness of ARMAS with real data while exposing areas of weakness in the ARMAS framework.

2.6.2 ARMAS-SOM Test with Simulated Sensors

Previous work [2] utilized simulated sensor data to further test the capabilities of ARMAS, and more specifically SOM. This work demonstrated ARMAS-SOM's capabilities in (1) detecting various faulty sensors and (2) determining when to pull in collaborative sensor information to maintain observability. In the case that not enough sensor information was provided, additional ranging sensors, which report the range between a main unmanned aerial vehicle (UAV) and another UAV, were added.

The main simulation which tested the ARMAS-SOM framework with simulated all-source data was 27 minutes long and consisted of four separate regions, where a team of five optionally coordinating vehicles would pass (see Figure 9), indicated by

the following:

1. Permissive (0-5km)
2. L1 Jamming (5-50km)
3. L1 Jamming + L2 Spoofing (50km-100km)
4. Permissive (100km - End of Simulation)

The motivation for the different regions was to test ARMAS-SOM's FDE capabilities when L1 is jammed and L2 is spoofed, and the capabilities in detecting a threat to resilience [2], in particular when faulty sensors are removed. This collaborative result is then compared to the result produced by a non-collaborative instance of ARMAS-SOM, revealing that SOM is able to effectively determine when to add more sensor information by adding in collaborative-ranging measurements and that the additional information greatly improves the size of the guaranteed position zone (see Figure 10) in comparison to the non-collaborative instance of ARMAS. This demonstration also shows that the additional layer ensures "guaranteed resilience to a single simultaneous sensor failure for 81 percent of the total simulation time versus 5 percent for a non-collaborative ARMAS instance" [2]. In other words, this research demonstrates the effectiveness of ARMAS-SOM in maintaining sufficient sensor information in order to remove a faulty sensor in the case of a single sensor failure at a given time.

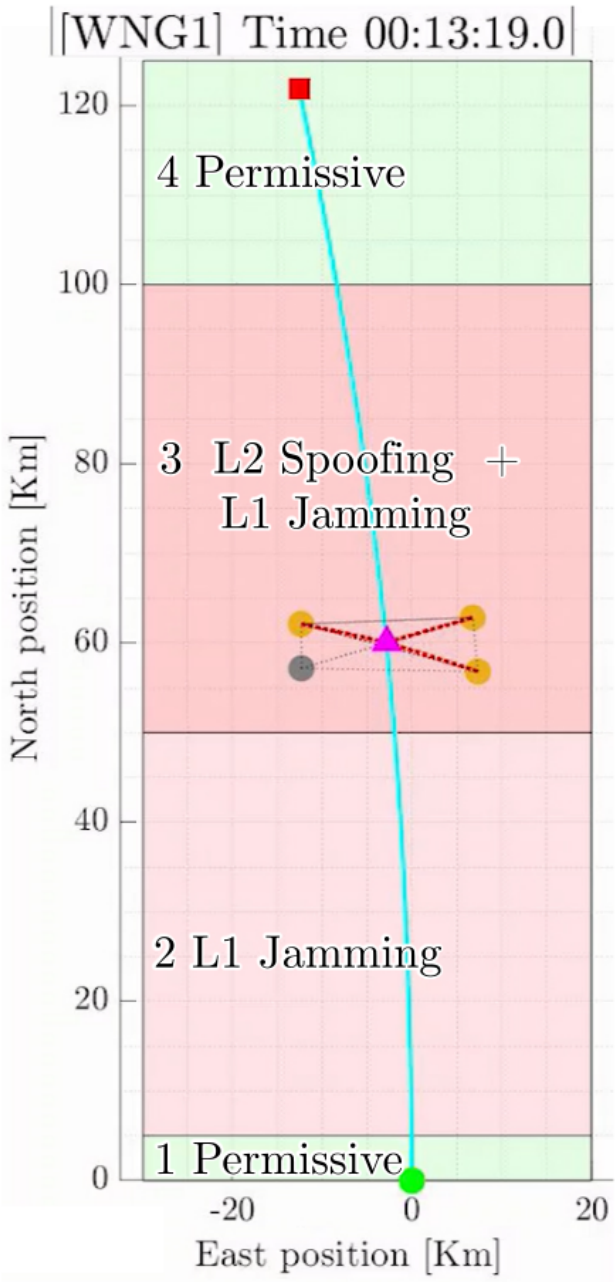

Figure 9: The four regions within Gipson's work as indicated by each number. The UAV (indicated by magenta triangle) began flight at bottom and travelled along the cyan colored path [2]

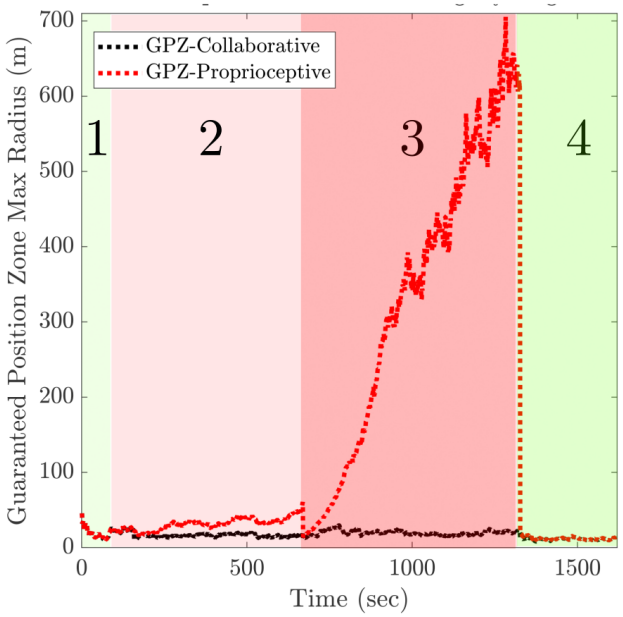

Figure 10: The Guaranteed Position Zone (GPZ) for the collaborative and proprioceptive solutions compared, demonstrating the effectiveness of collaboration [2].

2.7 Background Summary

The purpose of the background section was to demonstrate the need for reliable, all-source navigation with integrity, explore the current state of research in this area, and explain the focus of this thesis within this context. All-source navigation was demonstrated to be an increasingly effective way of providing a more accurate navigation solution, especially in the event of having little to no GNSS data, by fusing sensor information from multiple sources. However, adding more sensors introduces more opportunities for faulty sensor data negatively affecting the navigation solution, making a framework such as ARMAS necessary. The ARMAS-SOM framework fuses all-source sensor information, monitors the integrity of the system, excludes faulty sensors, and helps maintain an accurate and resilient navigation solution. Previous research has tested ARMAS with various simulated sensors and with real-life L1 pseudorange data, however, this thesis adds onto this research by testing additional real flight data (i.e. ranging radios and some vision-based position data) and by taking in asynchronous sensor data. In addition to testing ARMAS-SOM with real data, three main measurement metrics: RSS error, NEES, and GPZ, are employed to measure the reliability of the estimator and show the effectiveness of collaboration. This brings us closer to the end goal of implementing the ARMAS-SOM framework to reliably provide a navigation solution with reliability and integrity in a real-life and real-time situation.

III. Methodology

This chapter first details the testing completed during this project, beginning with the experiment setup at Camp Atterbury, IN. Afterwards, the data post-processing methods will be explained for each sensor type. Finally, the procedure for evaluating the Autonomous and Resilient Management of All-Source Sensors (ARMAS)-Stable Observability Monitoring (SOM) framework using the test data will be explained, taking note of the similarities and differences between this and previous research.

3.1 Flight Test

Evaluating ARMAS-SOM with real data requires real-world tests that provide data from multiple types of alternative navigation sensors. In the summer of 2021, the Air Force Institute of Technology (AFIT) Autonomy and Navigation Technology (ANT) Center was working to execute a series of flight tests to prepare for a small flight demonstration of alternative navigation technologies at the Army PNTAX exercise. We were able to add a set of extra sensors and some extra test objects to several of the planned flights to obtain data for this work. This section describes the flight test experiment, the software, and the hardware necessary to acquire the data.

3.1.1 Experiment Design

There were a number of main objectives this flight experiment sought to accomplish. The primary objectives included setting up an experiment with two vehicles while collecting timestamped GPS and ranging radio data for each to test ARMAS-SOM with real data. An overview of the experiment design created to accomplish these objectives will now be explained.

The overall experiment design involved two mobile vehicles: an aircraft as the

main vehicle and a car as a secondary vehicle, each equipped with a ranging radio. During the test, Global Navigation Satellite System (GNSS) data was collected for both the aircraft and for the car. In addition to the two ranging radios onboard the aircraft and the car, two stationary units were placed on tripods, approximately 2 meters above the ground. One of these radios was located just off the south end of the runway and the second was near the center of the runway, indicated by green dots in Figure 11. The radios will be identified throughout as follows: radio 1 (ID 100) on board the aircraft, radio 2 (ID 101) on board the car, radio 3 (ID 102) near the center of the runway, and radio 4 (ID 103) just south of the runway. Figure 11 shows the repeated path of the car during the duration of the flight (blue path), the two stationary P440 unit locations (green dots), and the origin location (red rectangle).

The origin location is where the aircraft's flight was initiated and where the East-North-Up (ENU) coordinates are centered (Lat: 39.346186 Lon: -86.009495 Alt: 187.02439). The flight path of the aircraft is not pictured in Figure 11, but the aircraft flew in various flight patterns near the coordinates shown in this same Figure.

3.1.2 Initial Software Setup

Keeping in mind the general idea of the setup, each component of the software which is relevant to the data for this thesis, will now be explained. This will be done by giving an overview of the software onboard the 2 mobile vehicles: the aircraft and the car. Then the software used to upload the proper settings to the two stationary ranging radio units will be explained.

3.1.2.1 Aircraft Software

The software on-board the aircraft records various data points by default. This includes the GNSS data and vision data (utilized for this thesis), and other data such

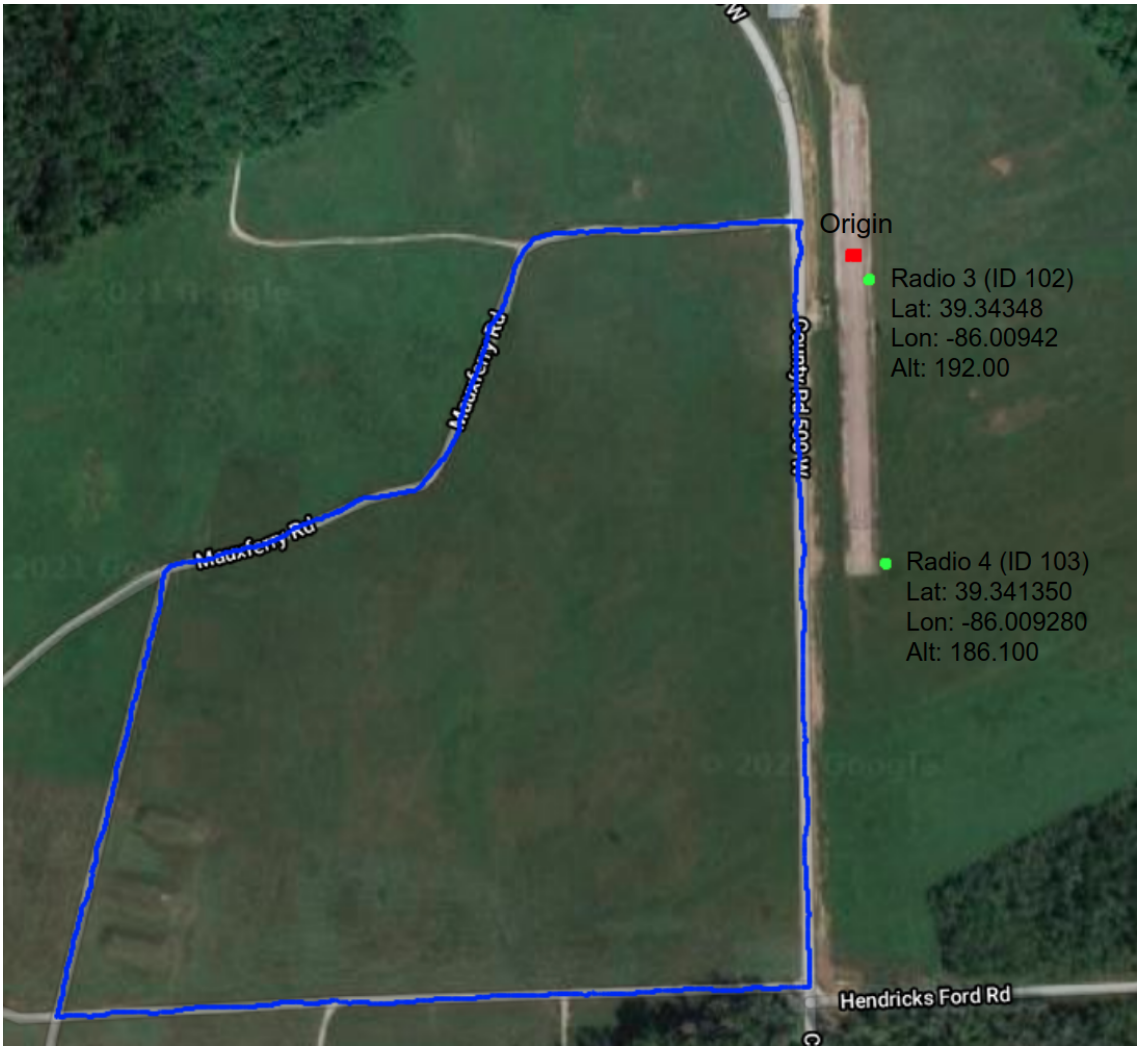

Figure 11: Top view of the runway and car path during data collection. Blue indicates path of car, red square is the origin, and the green dots represent the two stationary ranging radio locations.

as inertial measurement unit (IMU) and lidar data (not in this thesis). The aircraft is also equipped with a Precision Time Protocol (PTP) server, operating on Global Positioning System (GPS) time, allowing for the assumption of synchronized clocks between the unmanned aerial vehicle (UAV) and car. The main additional software added to the aircraft is the P440 Driver. The main goals of this P440 driver is to: (1) upload the proper settings to the P440 and (2) record the ranging data into a log file.

The P440 radios operated in RangeNet Time Division Multiple Access (TDMA) mode, as detailed in section 2.3.3. This means that the radios operate in a network with their clocks synchronized to the master radio. This mode requires the user to define a "slot map" which is a table enabling the user to define specific time slots at which two radios perform a range conversation and the settings that correspond to this range conversation.

The master radio is the radio which is defined as the "requester" radio in the first time slot. For this reason, the first slot is identical for all of the radios in the network. The remaining slots are complementary from radio to radio, but not identical. This is because at each given time slot (except the first time slot), two ranging conversations occur at the same time. For example, in time slot three, radio one can range with radio four on channel zero, while radio two and radio three range on channel one. This requires slot three to be the same on radios two and three (with the exception of the antenna polarization), but different from that of radios one and four.

Slot #	Req ID	Res ID	PII	Code Channel	Antenna Mode	Slot Type	Flags	Slot Duration (ms)
0	100	101	7	0	0	0x0	1	30
1	100	102	7	0	0	0x0	1	30
2	100	103	7	0	0	0x0	1	30
3	100	101	7	0	0	0x0	1	30
4	100	102	7	0	0	0x0	1	30
5	100	103	7	0	0	0x0	1	30
6	100	101	7	0	1	0x0	1	30
7	100	102	7	0	1	0x0	1	30
8	100	103	7	0	1	0x0	1	30
9	100	101	7	0	1	0x0	1	30
10	100	102	7	0	1	0x0	1	30
11	100	103	7	0	1	0x0	1	30

Table 2: Slot map uploaded to main P440 radio on the aircraft

For the full slot map definition uploaded to radio one (ID 100) on-board the plane, refer to Table 2. To understand the meaning of the slot map, each column will briefly be described. The first column refers to the slot number, beginning with zero and

moving up to the last slot. The next two columns refer to the requester radio ID and the responder radio ID. The requester is the radio which requests a range from another radio and the responder radio responds with a ranging packet including the ranging information. Column four refers to the integration index. Changing the integration index modifies the number of pulses each radio symbol uses. In short, increasing this value increases the max range for a range conversation but it also increases the length of time for that range conversation.

Column five contains the code channel. This value refers to the channel on which the range conversation takes place. At any given time, a single range conversation may take place on the same channel and a single radio can only have one range conversation during a given slot. Nonetheless, if two separate radios converse on a different channel, then multiple range conversations may occur simultaneously.

Column six contains the antenna mode. Each P440 radio has two antennas which are oriented perpendicular to each other, corresponding to antenna mode A (0) and antenna mode B (1). In order to maximize the opportunity for a successful range conversation, the four possible combinations of antenna modes are alternated through for each slot map definition (see section 2.3.3.1 for more detail on the motivations for alternating pols). For example, the entire slot map consists of each radio ranging with the other radio alternating between antenna modes AA, AB, BA, and BB, where the first letter refers to the antenna mode of the requester and the second to the antenna mode of the responder. Column seven contains the slot type for this slot. This can be either a range slot (1) or a data slot (2). The "flags" parameter in column eight is for indicating whether or not the requester (bit 1) or responder (bit 2) radio will include a data packet with the range request or response. This functionality is not employed in this thesis, but will likely be used in future research to send the responder radio's position to the requester radio with each range response.

The last parameter in column nine is the slot duration time. If this is set to zero, the range conversation will take place for the minimum amount of time possible. This minimum time differs depending on the type of slot it is (range or data), and the Pulse Integration Index (PII) value. A higher PII value means a higher minimum slot duration. For the purposes of this thesis, a value of 30 ms was selected. The minimum time for this PII (7) could be about 22 ms, however, when this was used, the software was unable to collect ranging data between each ranging conversation. Increasing this time by 8ms allowed a range conversation to occur, the code to collect that data and then move onto the next slot. Please note that this slot map in Table 2 is only for radio 1 (ID 100) on-board the aircraft. See Table 3 for radio 2's slot map.

3.1.2.2 Car Software

The car software consists of three main components relevant to this thesis: (1) P440 driver, (2) Swift Console, and (3) PTP Server. The P440 driver is almost identical to that on-board the plane. The main difference is the slot map is not identical. The slot map contains the same number of slots, but contains a complementary set of slot map definitions. For reference see Table 3.

Slot #	Req ID	Res ID	PII	Code Channel	Antenna Mode	Slot Type	Flags	Slot Duration (ms)
0	100	101	7	0	0	0x0	1	30
1	101	103	7	1	0	0x0	1	30
2	101	102	7	1	0	0x0	1	30
3	100	101	7	0	1	0x0	1	30
4	101	103	7	1	1	0x0	1	30
5	101	102	7	1	1	0x0	1	30
6	100	101	7	0	0	0x0	1	30
7	101	103	7	1	0	0x0	1	30
8	101	102	7	1	0	0x0	1	30
9	100	101	7	0	1	0x0	1	30
10	101	103	7	1	1	0x0	1	30
11	101	102	7	1	1	0x0	1	30

Table 3: Slot map uploaded to the P440 radio on the car

The swift console corresponds to the swift piksi GNSS receiver. The swift console constantly runs during the flight test and records GNSS data to a json file. This file is then converted to a csv file containing latitude-longitude-altitude (LLA) coordinates of the car at a given timestamp. This is the data which is used for the truth position of the car at each timestamp. The ranging radio data is post-processed to determine the GNSS location of the car at the time of a given range.

Since the radios are ranging together on separate vehicles, there is also need for a PTP server on-board the car to ensure the clock matches up with the clock on-board the plane. To accomplish this, an identical PTP driver to that on the plane continually runs to ensure the clock running the P440 driver on the car is as close to GPS time as possible.

3.1.2.3 Stationary Ranging Radio Software

The operation of the stationary P440 radios differed from the other two units because these radios were not connected to a computer during the flight test. For this reason, these radios only required an initial uploading of the settings and then were left alone. The specific slot maps for radio three (ID 102) and radio four (ID 103) are not pictured here, but correspond closely to the slot maps of radios one and two (Tables 2 and 3). For the slots in which these two radios (103 and 104) are not shown in the slot maps pictured above, they are ranging with each other. The most important observations to note between the four different slot maps is that (1) the first slot (slot 0) is identical for all radios because this slot establishes the master radio and (2) the rest of the slots differ from radio to radio, but do not contradict each other, rather they are complementary to allow for highest efficiency on the network.

3.2 Data Post-Processing

The main data collection process has already been described, now the post-processing process for each portion of the data will be described, beginning with a brief description of the measurement processors contained in ARMAS. After this, each data type will be considered individually. This will begin by explaining the GPS truth data recorded on the plane and the car. Then the rest of the data: L1 satellite data, ranging radio data, vision-based position data, and the simulated velocity data will be described. This data is what is actually used to evaluate the ARMAS framework.

3.2.1 ARMAS Measurement Processors

The ARMAS-SOM framework relies on several types of measurement processors to process each of the all-source navigation sensors. These measurement processors are used to process information from different sensors, meaning that each of the different types of sensors (i.e. velocity, ranging, position, etc.) have a different measurement processor. The four measurement processors utilized in this thesis, and which will be explained here, process information from four types of sensors: pseudorange sensors (e.g. L1 and L2 sat data), ranging sensors (e.g. P440 radios), position sensors (e.g. vision data, MagNav data, etc.), and velocity sensors. In the ARMAS framework, each sensor is added as a specific type of measurement processor and the data for that sensor is updated as it is received.

3.2.2 GPS Truth Location Data

The sections for the GPS data are separated into two sections: (1) GPS truth location data (this section) and (2) L1 and L2 satellite data. This section refers to the truth data which recorded the GNSS position estimate for the aircraft at a

given time. More specifically, this data includes the timestamped LLA data for the aircraft. These coordinates were then converted into ENU coordinates centered on the position of the aircraft at the beginning of the flight. The GPS location data for the plane was utilized for two main purposes: (1) the GPS truth location and (2) to create a simulated velocity sensor. For this reason, even when the L1 satellite sensors are removed, the filter solution still relies on the GPS data (with added noise) for the velocity sensor. See Table 4 to see an example data matrix which contains GPS location data, taking note that a "nan" value indicates no data at that particular timestamp.

Time	East	North	Up
0	0	0	0
0.05141	1.0001	5.4421	2.2451
0.8245	2.7781	4.9214	3.1281
1	3.0532	5.1532	3.3356

Table 4: Example of data included with GPS Truth Location

3.2.3 L1 Satellite Sensor Data

In ARMAS, each L1 satellite is added as a separate sensor. For example, if there are 8 satellites, as with this data, there will be 8 L1 satellite sensors. If L2 satellite data was also used, then there would be 16 satellite sensors, 8 L1 and 8 L2 sensors. For the purposes of this thesis, the satellite data is recorded in a series of 6 columns. The first column contains the timestamp. Column 2 contains the L1 Pseudorange (PR), column 3 contains the L2 PR (although not actually used), and columns 4-6 contain the satellite position at that timestamp. All of this data was calculated using the ephemeris and observation data from the rinex data files. See Table 5 for the example data format for a single L1 and L2 satellite, which corresponds to two ARMAS sensors, one for L1 and one for L2. In this thesis, the L2 PR is yet to be

implemented, hence why all the values in the L2 column are “nan”.

Time	L1 PR (m)	L2 PR (m)	East (m)	North (m)	Up (m)
0	22301616.96	nan	-14146978.39	-11713571.23	12865696.44
0.05141	nan	nan	nan	nan	nan
0.8245	nan	nan	nan	nan	nan
1	22302096.53	nan	-14146954.64	-11716312.68	12864060.12

Table 5: Example of L1 Satellite data

The L1 pseudoranges and SV positions were calculated by using code provided from previous research [3]. This code follows typical procedures of modifying the raw PRs to account for various errors and calculating satellite positions at specific times using ephemeris data.

3.2.4 Ranging Radio Sensor Data

The ranging radio data contains two main components with timestamps: (1) the radio range and (2) the radio location in ENU coordinates at each timestamp. This data is organized into a series of 5 columns. The first column, as with every other sensor, is the timestamp. The next three columns contain the ENU coordinates of the radio and the last column the range recorded at this timestamp. See Table 6 for the example data format for ranges between two radios.

Time	East (m)	North (m)	Up (m)	PRM (m)
0	nan	nan	nan	nan
0.05141	nan	nan	nan	nan
0.8245	6.440240076	-15.38023396	4.9753595	14.791
1	nan	nan	nan	nan

Table 6: Example of data format for ranging radio data

Due to the nature of the Kalman filter, each measurement requires a covariance value for each measurement. For the case of the ranging radios, this covariance value

was determined by comparing the difference between the ranging radio data to the range calculated using truth GPS location. Plots showing the comparison between the actual ranging data vs. the simulated ranges are seen in Figures 12 to 14. As you can see visually from the plots of the difference between the ranging data and the simulated values, they match up very similarly with similar covariance and mean values. The covariance of the radios ranged from 3.32^2 to 4.58^2, while the means ranged from -1.6680 to 1.0754. The average covariance between the three radios was 4.0577^2, while the average mean value was -0.0998.

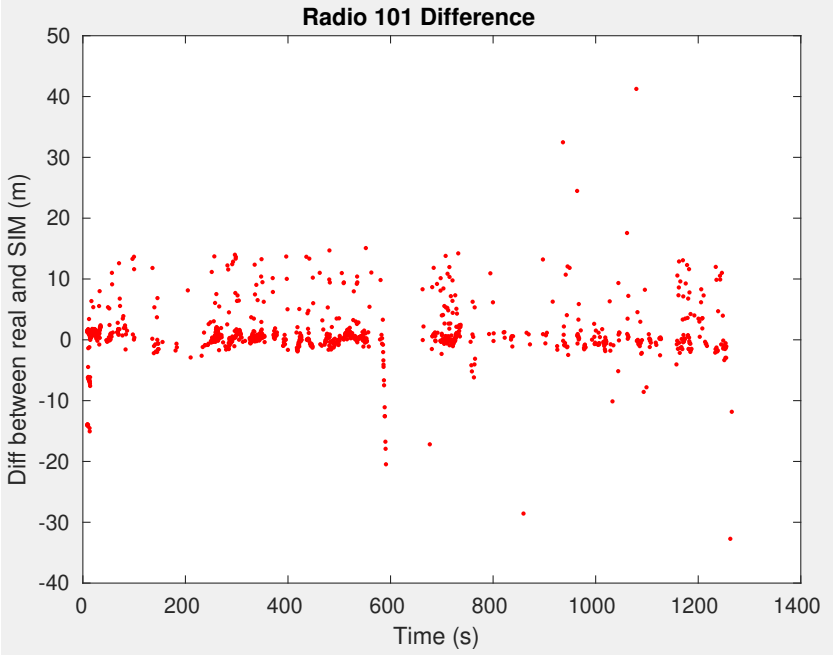

Figure 12: Difference between simulated and actual ranges between main radio and radio 101 ($\mu = 1.0754$ and $\sigma = 4.5846$)

Through observing difference plots in Figures 12 to 14 there are a number of important limitations to consider. First, each plot has a general range of about 5 meters where the difference fluctuates up and down. This fluctuation indicates that the ranging radios are providing more precise information than the GPS truth data being used for the simulated ranges. Using more precise GPS data in the future (i.e. differential GPS) could help give a better idea of the actual error and distribution of

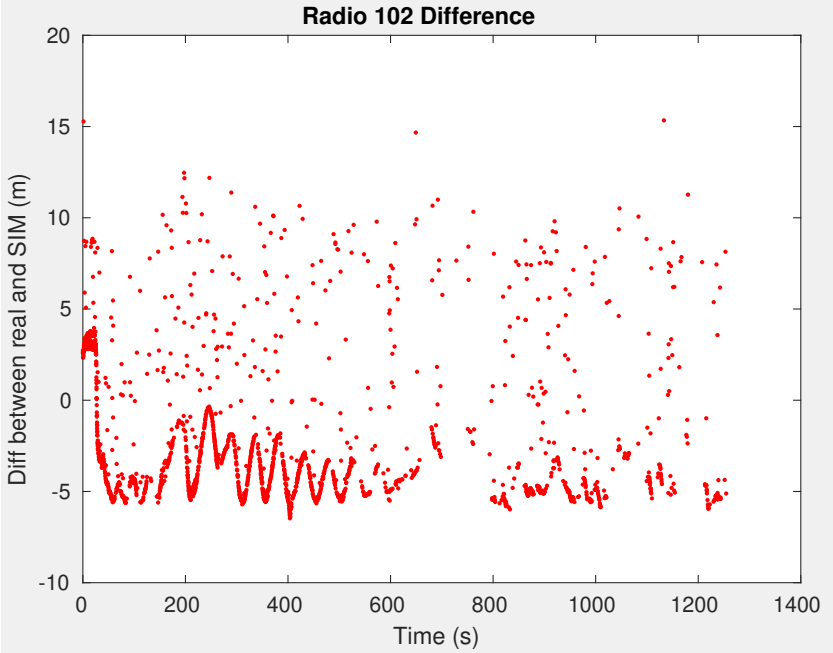

Figure 13: Difference between simulated and actual ranges between main radio and radio 102 ($\mu = -1.6680$ and $\sigma = 4.1643$)

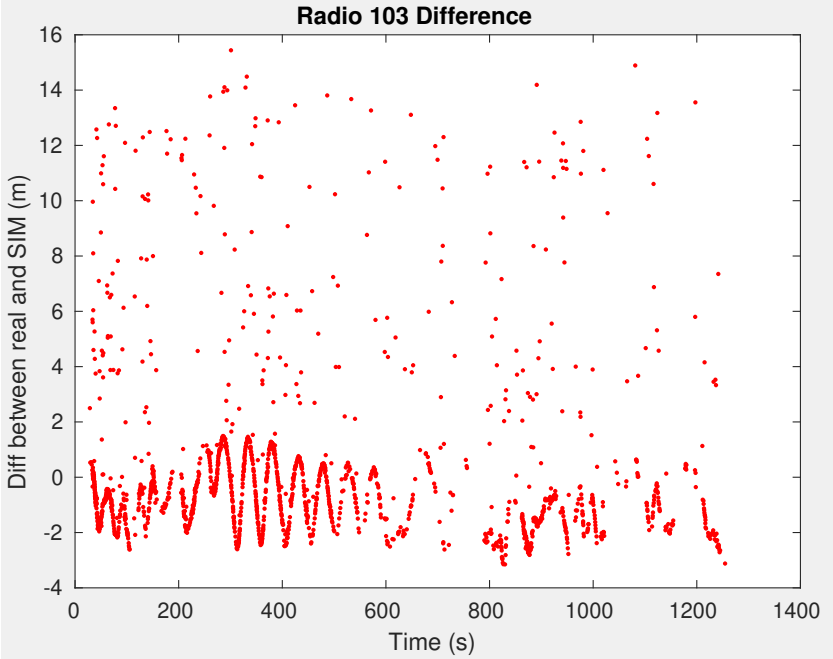

Figure 14: Difference between simulated and actual ranges between main radio and radio 103 ($\mu = 0.2931$ and $\sigma = 3.3220$)

these radios.

The other, and probably more noticeable, effect throughout each test scenario run is that intermixed within this general range of values, each radio has some higher error values dispersed throughout the flight. It is not uncommon to see these values go 10 meters away from the general error data-set and some even reach up to 30 meters in the case of radio 101.

To illustrate just how much these data points affect the data, let us consider what the covariance and mean values for this data would be if only the main cluster of data was used (see blue dots in Figures 15 to 17). In this case the mean values move from 1.0754, -1.6680, and 0.02931 to 0.2823, -4.0297, and -0.8365. The covariance values change from 4.5846^2, 4.1642^2, and 3.3220^2 to 0.8773^2, 1.3320^2, and 1.0520^2. As you can see, this changes the average standard deviation from 4.0577 to 1.0871.

The blue data represents about 25 percent of the overall ranging data, which indicates that $\frac{1}{4}$ of the data is increasing the standard deviation by an average of three across the radios. Tests were run to see if cutting out these higher error values (i.e. using only the blue data) would improve the overall filter results, but based on this data set the benefits of cutting out the noisy data were overshadowed by the downsides. Nonetheless, it is evident that these outlier datapoints affect the overall distribution of the radio data and further consideration of these effects should be considered in the future.

It is also important to note that while this particular method of removing the noisy data points cannot be employed in real-time as it is, it represents what could be accomplished in the future via residual monitoring. In residual monitoring, if the residual value is higher than a certain user-defined threshold, then the filter would not be updated with the measurement. This accomplishes a similar result to manually removing the noisiest ranging radio values via a threshold prior to input

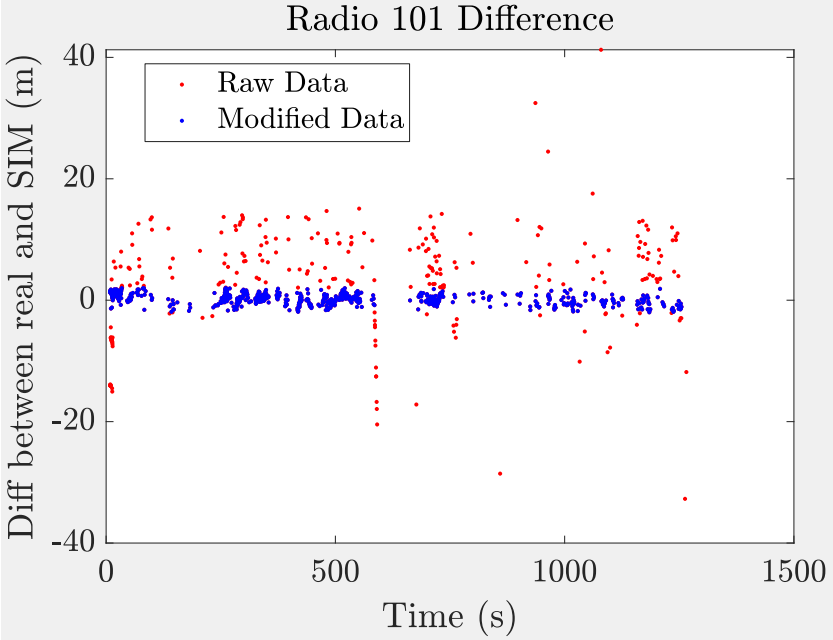

Figure 15: Comparison of real difference data (red and blue) vs. modified difference data (blue only) with noisy values removed for radio 101 ($\mu = 0.2823$ and $\sigma = 0.8773$)

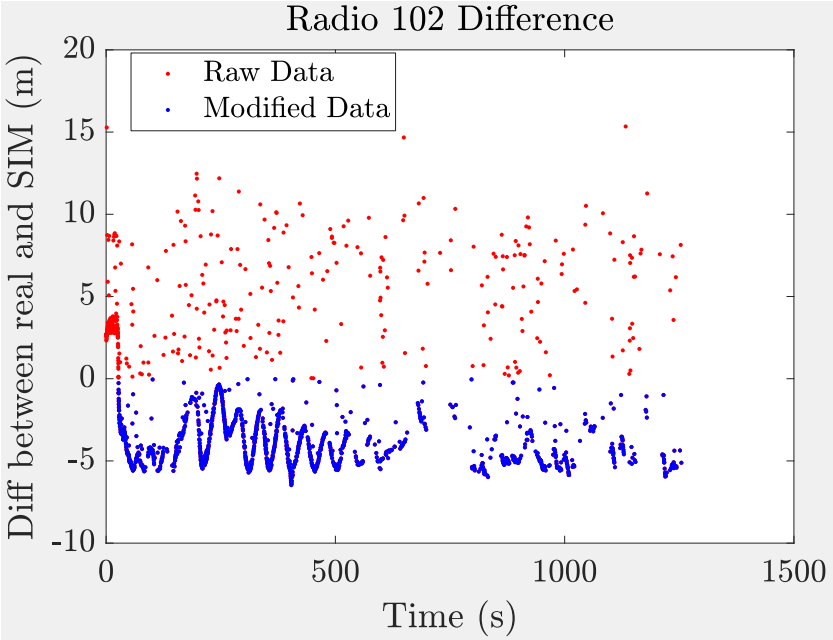

Figure 16: Comparison of real difference data (red and blue) vs. modified difference data (blue only) with noisy values removed for radio 102 ($\mu = -4.0297$ and $\sigma = 1.3320$)

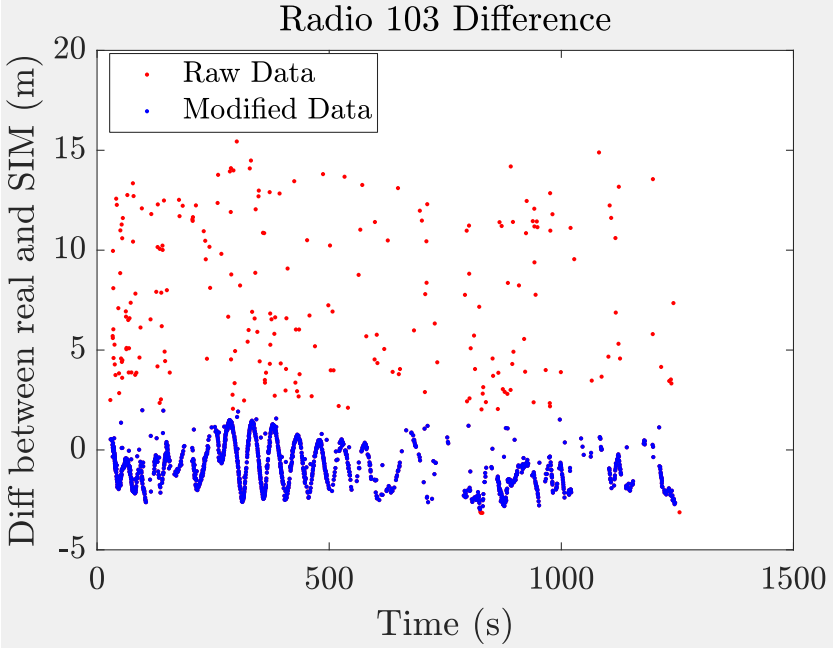

Figure 17: Comparison of real difference data (red and blue) vs. modified difference data (blue only) with noisy values removed for radio 103 ($\mu = 0.2931$ and $\sigma = -0.8365$)

into ARMAS. The main difference being residual monitoring may be accomplished in a real-life situation while the method used here cannot. In the end, the real radio ranging data produces good results, often better than the modified data. For this reason, it is not as urgent that we add the residual monitoring capability as originally thought.

3.2.5 Vision-Based Position Sensor Data

The last sensor which contained actual recorded data is the vision-based postion sensor. This sensor provides position estimates whenever the imaging system is able to utilize the terrain map to detect an estimated position. The data for this sensor consists of 4 columns, containing a timestamp followed by the estimated ENU coordinates at this time. Column 1 has the timestamp and columns 2-4 the ENU coordinates (see Table 7).

Time	East (m)	North (m)	Up (m)
0	nan	nan	nan
0.05141	3.226002837	10.45314458	-16.48958209
0.8245	nan	nan	nan
1	nan	nan	nan

Table 7: Example of data format for vision-based position sensor data

Due to there only being 44 successful image position matches during the entire duration of the flight, simulated data was added in with this real data. The covariance values for the real data was calculated and used to create some simulated position sensor data which matched the distribution of the real data. The covariance values ended up being 6.3741^2, 5.7113^2, and 7.7543^2 for the east, north and up directions respectively. These covariance values were used to add noise with the same covariance to the truth position data to produce enough simulated position data to have an update rate of about $\frac{1}{15}$Hz over periods where there is no real-life position sensor information provided by the camera. See Figure 18 comparing the real position data prior to adding the simulated data vs the real position data plus the simulated data. As you can see, there is an addition of 82 simulated data points, spread out over areas in which there were no image matches (indicated by blue). Figure 19 highlights the difference between the actual position of the aircraft and the positions given by the camera sensor. This also helps show the distributions are similar for the simulated camera data as with the 44 actual image matches.

3.2.6 Simulated Velocity Sensor Data

The final sensor, and only fully simulated sensor, is the velocity sensor. There were a couple motivations for adding this sensor: (1) to add another sensor to ARMAS in order to test ARMAS' capabilities in maintaining resiliency amid jammed and/or spoofed sensors, and (2) to closer match the sensors used by Gipson in his

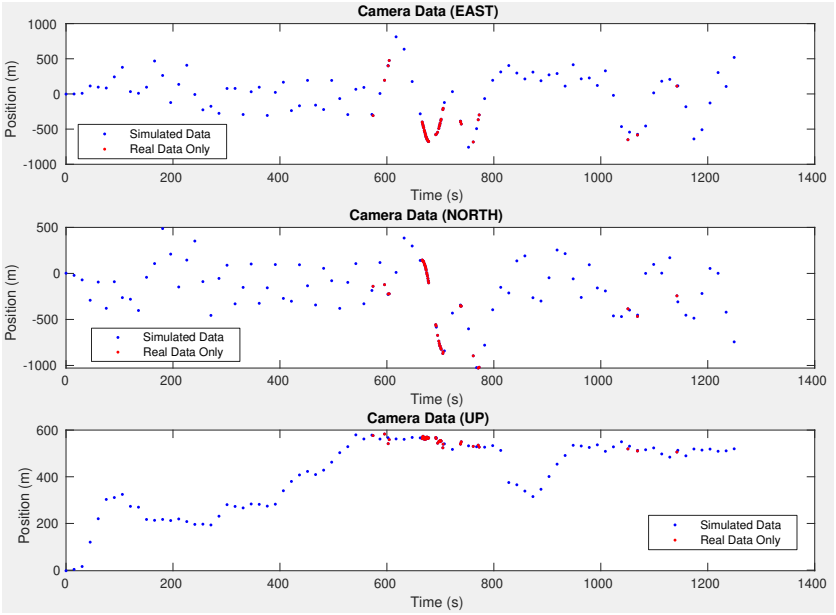

Figure 18: Simulated and Real camera data for the East, North, and Up coordinates

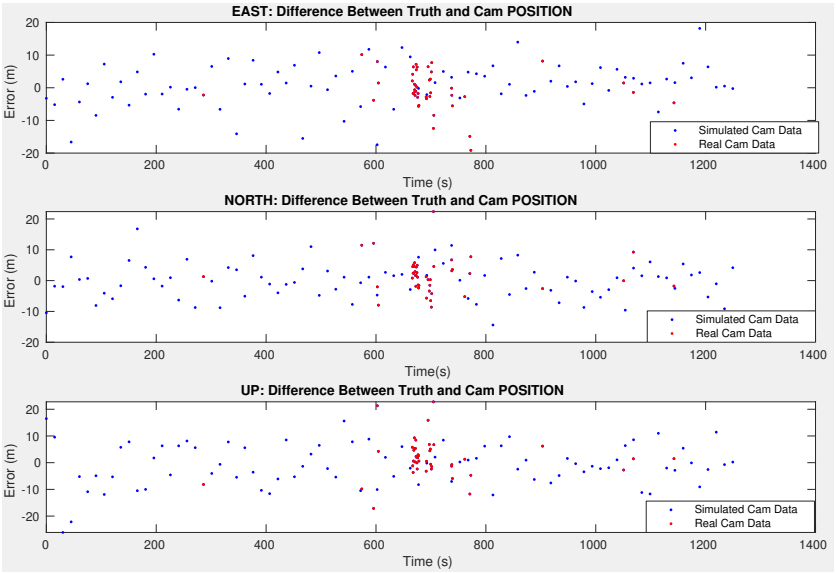

Figure 19: Difference between the truth position and the position provided by the camera sensor

papers [2][10]. The velocity data is organized as follows: the first column contains the timestamp and the last 3 columns contain the ENU velocity components. See table 8 for the example data format for this velocity sensor.

Time	East (m/s)	North (m/s)	Up (m/s)
0	-15.75868754	-21.87376741	-10.52635868
0.05141	nan	nan	nan
0.8245	nan	nan	nan
1	-9.67700403	8.901317561	1.772897246

Table 8: Example of data format for velocity sensor data.

3.3 Organizing the Sensor Data

As seen in the previous subsection, each sensor has its data timestamped, and arranged in a particular data format. The next requirement was to find a way to organize all of this data to evaluate the ARMAS-SOM framework as if it was happening in real-time. This required a method different from previous research.

There are three main areas of distinction between this research and past research [2][3], each of which will be explored here in brief. These include (1) the use of real data for L1, ranging radio, and imaging data, (2) accommodating for asynchronous data updates, and (3) accommodating for times in which a sensor did not receive an update. Numbers (2) and (3) will be linked together in a single section as the solution for one is closely tied to the other.

3.3.1 Real-Life Sensor Data

One of the main goals of this research is to test ARMAS-SOM with more sensor data and determine the strengths and weaknesses of ARMAS in dealing with this data. The advantage of simulated data is it requires no additional flight tests and can be easily created and tested. Yet, simulated data never fully represents actual data. Utilizing real data further validates the effectiveness of ARMAS and gives more confidence that it would be a reliable system to employ in real-time situations.

3.3.2 Accommodating for Asynchronous Data

The last two points (i.e. accommodating for asynchronous data and times at which sensors did not receive an update) are closely related to the first point. In other words, bringing in real data with multiple sensors introduces asynchronous measurements and ensures that we will have times when certain sensors are or are not reporting data. In order to accommodate for these two alterations, a "master data matrix" was created to store all the sensor information. This master data matrix organizes the sensors in their corresponding data formats from section 3.2 into one big, timestamped data matrix.

3.3.2.1 Data Matrix Structure

The data matrix is a 2-Dimensional matrix in Matlab which contains all the information for all the sensors at a given time. The rows of the data matrix are organized in ascending order of time in which the measurements occur, with each sensor's data designated to a certain set of column indices. In other words, each row indicates a specific timestamp, the timestamp in which any of the sensors reported a measurement; and each sensor is assigned to report its data to a predefined set of column indices. This data matrix organizes the data described in section 3.2 and its subsections into a single matrix.

Before moving on, it is important to note that in Table 9, "L1 Data" represents the 5 columns of data for the GPS satellite as described in section 3.2.3, "Rad Data" represents the ranging radio data described in section 3.2.4, "ENU Truth" represents the truth GPS data described in section 3.2.2, "Vel Data" represents the velocity sensor data described in section 3.2.6, "Pos Data" represents the position sensor data described in section 3.2.5, and "nan" means that there is no data for this sensor at this timestamp. Since adding in all the data for each sensor would not fit on a single

page, these labels are used to make it fit.

As an example, let us look at Table 9. Please note that the first column is the timestamp column. This column is used as the time variable within ARMAS and the filter propagates through this time column. The rest of the columns contain sensor data information. For example, columns 2-6 contain sensor information for a single GPS satellite. Columns 2-3 contain L1 and L2 pseudorange data respectively, and columns 4-6 contain the ENU coordinates of the satellite at this time. This is the same way in which the data was organized in section 3.2.3. The next GPS satellite is designated to the next 5 columns, 7-11. It follows an identical structure as the first satellite, 7-8 containing the L1 and L2 pseudorange data and columns 9-11 the ENU coordinates of the satellite at this particular time. This continues until the last satellite data is included. In this particular case, the satellite data ends at column 41 and then the ranging radio data begins (see Table 9).

Time	L1 SAT 3 (cols 2-6)	L1 SAT 4 (cols 7-11)	...	L1 SAT 31 (cols 37-41)	P440 101 (cols 42-45)	P440 102 (cols 46-49)	P440 103 (cols 50-53)	GPS Truth (cols 54-56)	Vel Sensor (cols 57-59)	Pos Sensor (cols 60-62)
0	L1 Data	L1 Data	...	L1 Data	nan	nan	nan	ENU Truth	Vel Data	nan
0.05141	nan	nan	...	nan	nan	nan	nan	ENU Truth	nan	Pos Data
0.8245	nan	nan	...	nan	Rad Data	nan	nan	ENU Truth	nan	nan
1	L1 Data	L1 Data	...	L1 Data	nan	nan	nan	ENU Truth	Vel Data	nan

Table 9: Example of the master data matrix

The ranging radios occupy 4 columns each with the first 3 columns pertaining to the ENU coordinates of the radio, and the last column containing the Precision Range Measurement (PRM) at this timestamp. Since there are 3 ranging radio sensors, columns 42-53 contain all of this data. After the ranging radio data, the GPS truth coordinates of the aircraft are recorded. These are the coordinates which are compared with the navigation solution estimated by the ARMAS framework. The next three columns contain the velocity sensor data. Each column corresponds to the ENU velocity coordinates respectively. The final 3 columns contain the vision-based position sensor's data measurements.

The point of organizing the data as so is not that the particular column order matters, but that it is a way of organizing the data by timestamp so that the code may easily detect what sensors did and did not report updates at a particular timestamp. This data matrix accommodates for asynchronous measurements because it is reporting data as it comes in from the sensors and it accommodates for sensors not reporting data at a given timestamp by including a "nan" value in that sensor's columns at that particular time. The ARMAS code will propagate through each timestamp in column 1, and at each timestamp it will determine if the sensor has reported data, by looking for a "nan" value or not, and update the filter's measurement accordingly. If the sensor reported a measurement at that timestamp, it will update the filter with that measurement. If the sensor did not, represented by a "nan" value, then it will not update that measurement until it reaches the timestamp where that sensor did receive an update.

3.4 Matlab Scenarios Overview

Various scenarios were tested with the data from the same flight test. Each test scenario differs in regard to the sensors used and when/if a bias was added to a sensor. The main goals of these various test scenarios are to display the effectiveness of ranging radios as additional all-source sensors and demonstrate the capabilities of ARMAS in detecting a faulty sensor while maintaining resiliency. The overall duration of each test scenario is 1277 seconds and the details regarding each test scenario is described within Chapter IV.

3.5 Methodology Summary

This research builds upon [2] by introducing real GPS, ranging radio, and some vision-based position data. This real data was collected during flight tests at Camp

Atterbury, IN in which there were two mobile vehicles (aircraft and car), each containing a ranging radio, and two stationary ranging radio units on the ground. The main way this system accommodates the use of real data is through utilizing the "master data matrix", which contains the data for each sensor at each time. By allowing for the system to update a measurement based upon the whether or not there was a measurement update at that particular time, this updated system demonstrates that the ARMAS framework could be applied in real-time when asynchronous measurements are occurring regularly. The various test scenarios are conducted with the goal of extensively testing ARMAS-SOM's abilities to exclude faulty sensors and add additional sensor information when threats to resiliency are detected. Through the comparison of the test scenarios where all the sensors are available for the duration of the flight with the test scenarios when L1 sensors experience jamming and/or spoofing, ARMAS-SOM's capabilities are stretched in order to see when/if the framework breaks down. Overall, this setup is intended to: (1) demonstrate ARMAS is able to operate with real life sensors including L1 satellites and ranging radios, and (2) exhibit the strengths and weaknesses of the ARMAS-SOM framework in these conditions.

IV. Results and Analysis

4.1 Results and Analysis Overview

This Chapter will look at each Test Scenario (TS) and analyze the results of each when tested with real data. This will be done by taking a look at each of the relevant measurement metrics described in Section 2.5 for each scenario and outline the strengths and weaknesses of the overall Autonomous and Resilient Management of All-Source Sensors (ARMAS)-Stable Observability Monitoring (SOM)framework throughout. Each Test Scenario is briefly described, followed by an analysis and comparison of the results between the different scenarios.

4.2 Matlab Scenarios Results and Analysis

This section will briefly describe the simulations which were conducted and present comparative results with the intention of demonstrating the effectiveness of the ranging radios and other real sensors. Unless otherwise indicated, the ranging radio sensor data used is the real radio ranges without any bias added or data points removed.

4.2.1 Test Scenario 1: L1, Vel, and Pos Sensors

The first test scenario consists of the following sensors for the entire duration of the flight (1277 seconds):

1. 8 L1 Satellites
2. 1 Velocity Sensor
3. 1 Position Sensor

The first two test scenarios are unique in that they both keep the same exact sensors for the entire duration of the flight. This first test scenario tests ARMAS-SOM when L1, velocity, and position sensors are available for the entire duration of the flight. This is meant mainly for comparison to the other test scenarios with this test scenario representing the best possible result.

4.2.2 Test Scenario 2: L1, Vel, Pos, and Ranging Sensors

The second test scenario consists of the following sensors for the entire duration of the flight (1277 seconds):

1. 8 L1 Satellites
2. 1 Velocity Sensor
3. 1 Position Sensor
4. 3 Ranging Radio Sensors

This test scenario differs from test scenario 1 only in the addition of the 3 ranging radio sensors for the entire duration of the flight. See the following sections for a comparison between this test scenario and that of test scenario 1.

4.2.3 Comparison: Test Scenario 1 and 2

Before moving to test scenario 3, a comparison of test scenarios 1 and 2 will be considered. In this data, TS2 refers to Test Scenario 2 and TS1 refers to Test Scenario 1. This comparison highlights the effects of the ranging radios over the duration of the entire data set when the ranging radios are not necessary for maintaining observability. This data compares the Root Sum of Squares (RSS) error and Normalized Estimation Error Squared (NEES) reported only at the times in which the proprioceptive instance of ARMAS reports data. Since the collaborative instance has more sensors (i.e. the

ranging radios), the collaborative will also be reporting results at times in which the proprioceptive is not. For this reason, the plots are showing the collaborative results "at the time of the proprioceptive" to highlight how the additional ranging radio measurements affects the results.

4.2.3.1 RSS Error: TS1 and TS2 Comparison

The RSS error plot reveals nominal difference between TS1 and TS2. This makes sense, as there are eight L1 satellites available for the entire duration of this test, meaning that the RSS error should be very low the entire time. This addition of the ranging radios proves not to be advantageous in the case of the RSS errors, although it certainly does not hurt the RSS errors to add them.

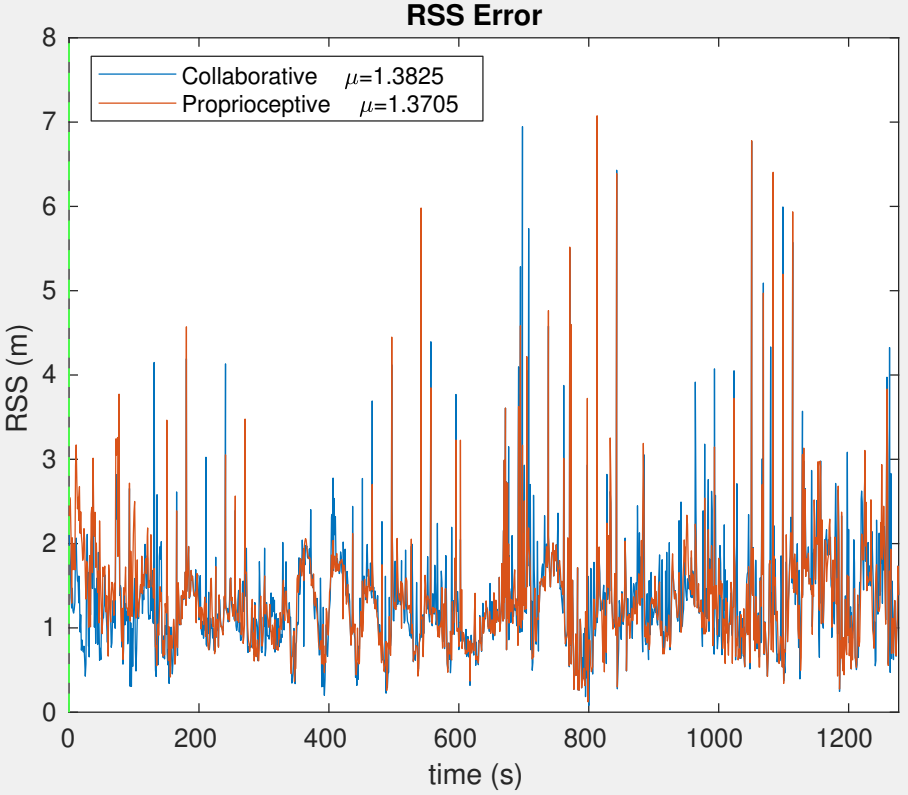

Figure 20: Comparison of the 3D RSS error for Test Scenario 1 and 2

4.2.3.2 NEES Results: TS1 and TS2 Comparison

The NEES, which measures the second moment of error, gives a good indication of whether or not the estimator is consistent. In this comparison between Test Scenarios 1 and 2 in Figure 21, TS2 produces a slightly better average NEES value ($\mu = 0.9704$) than that of test scenario 2 ($\mu = 0.9331$). This result indicates that the ranging radios improve the average NEES, but not by much. Again, this is consistent with what we would expect, as more measurements should raise the NEES.

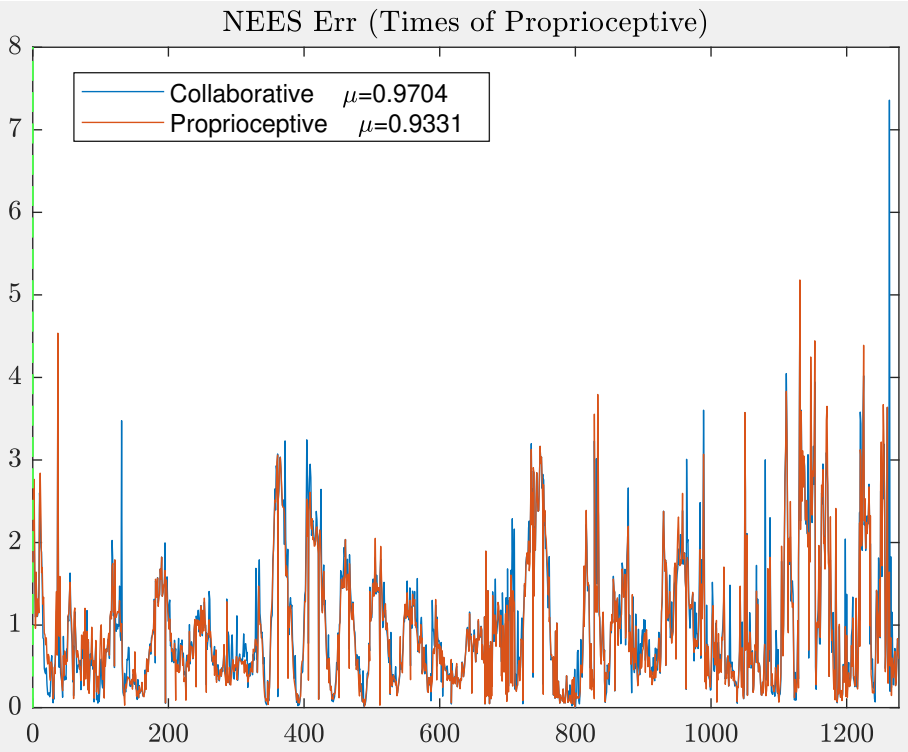

Figure 21: Comparison of the NEES for Test Scenario 1 and 2

4.2.3.3 SOM Results: TS1 and TS2 Comparison

Since Test Scenarios 1 and 2 do not contain any jamming or spoofing scenarios, it is no surprise that SOM is stable for the entire duration of the test run with the exception of some temporary unstable periods. These temporary unstable periods in

SOM are an artifact of using real data, and typically don't last for more than 1-2 seconds. The most common reason for a temporary unstable SOM reading is when one or more satellites have a temporary outage.

Overall, the comparison of Test Scenarios 1 and 2 reveals the ways in which the ranging radios impact the data when they are not necessary for maintaining observability. The mean values for the RSS and position errors do not differ in any statistically significant way, yet there is some improvement in the average NEES when the ranging radios are present. In light of these results, it is not advantageous to add the ranging radio sensors unless there is a threat to observability or there is an insufficient amount of satellite sensor data. Nonetheless, adding the ranging radios unnecessarily is much better than not having these additional sensors when they are needed (see Test Scenarios 3-6 for example).

4.2.4 Comparison: Test Scenario 2 Simulated vs. Real Ranges

Section 3.2.4 looked into the real ranging radio data which was fused in the ARMAS filter. While that section examined briefly the comparison between the real ranges and the simulated ranges, this section will further analyze the differences between the real radio ranges and the simulated ranges, and specifically how these differences impact ARMAS-SOM's performance. This section reveals how well the ranging radios could theoretically perform given the current data set and compares this to the real ranging radio data.

4.2.4.1 East-North-Up (ENU) Errors: Simulated vs. Real Ranges TS2

Figure 22 demonstrates that the covariance values are lower using simulated radio ranges than when real radio ranges are used. In Section 3.2.4, the analysis of the

real ranges showed a high percentage of outlier measurements and that the measurement errors were not Gaussian. It is not surprising then that in Figure 22, utilizing simulated ranges lowers the covariance (although not by much). The use of simulated ranging radios decreases the covariance values of the filter from $\sigma^2 = 1.1297^2$ to $\sigma^2 = 1.0067^2$, $\sigma^2 = 1.2972^2$ to $\sigma^2 = 1.1211^2$, and $\sigma^2 = 1.6590^2$ to $\sigma^2 = 1.1109^2$ for the ENU coordinates.

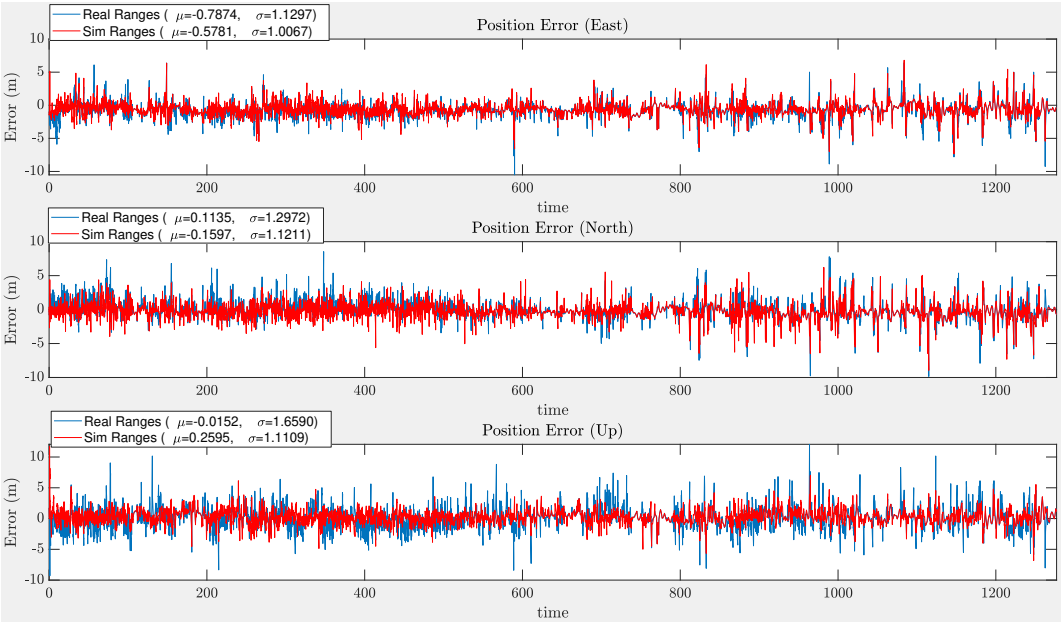

Figure 22: Comparison of the ENU errors for Test Scenario 2 using simulated ranges vs. real radio ranges

The simulated ranges have a controlled covariance of 1m (added in after the fact), while the real ranges have covariance values of $\sigma^2 = 4.5846^2$, $\sigma^2 = 4.1643^2$, and $\sigma^2 = 3.3220^2$, when the data are enveloped with Guassian curves. The simulated ranges produce only nominal improvements in the ENU errors, even though the measurements contain less noise.

4.2.4.2 NEES Results: Simulated vs. Real Ranges TS2

The average NEES is closer to one for the simulated ranges than for the real radio ranges. This is likely due to the lower covariance values for the simulated ranges, and the consistency of the simulated data. The average NEES being lower than one indicates that the covariance values for the ranges may be too large. Nonetheless, this average NEES does not indicate poor performance.

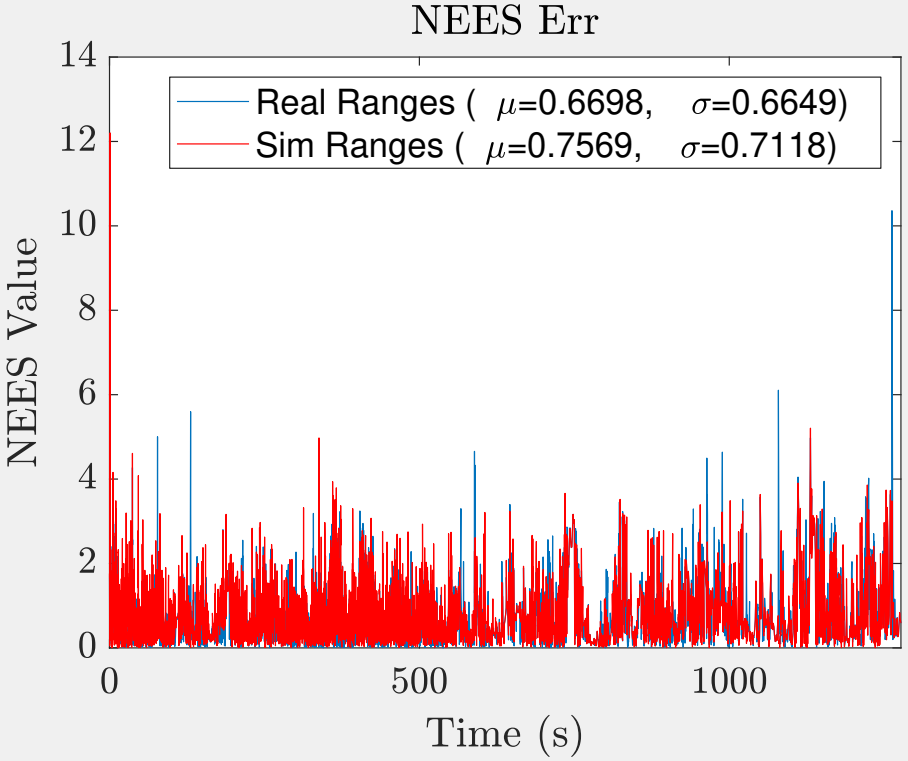

Figure 23: Comparison of the NEES error for Test Scenario 2 using simulated ranges vs. real radio ranges

4.2.4.3 Guaranteed Position Zone (GPZ) Results

The GPZ results reveal marginal, if any, improvements from using the simulated ranges to the real radio ranges. Since the GPZ is the union of the Fault Detection and Exclusion (FDE) layer results, we would not expect much difference between two tests

which employ the same sensors producing measurements at the same timestamps and contain L1 measurements for the entire duration. This is confrimed by the results shown in Figure 24.

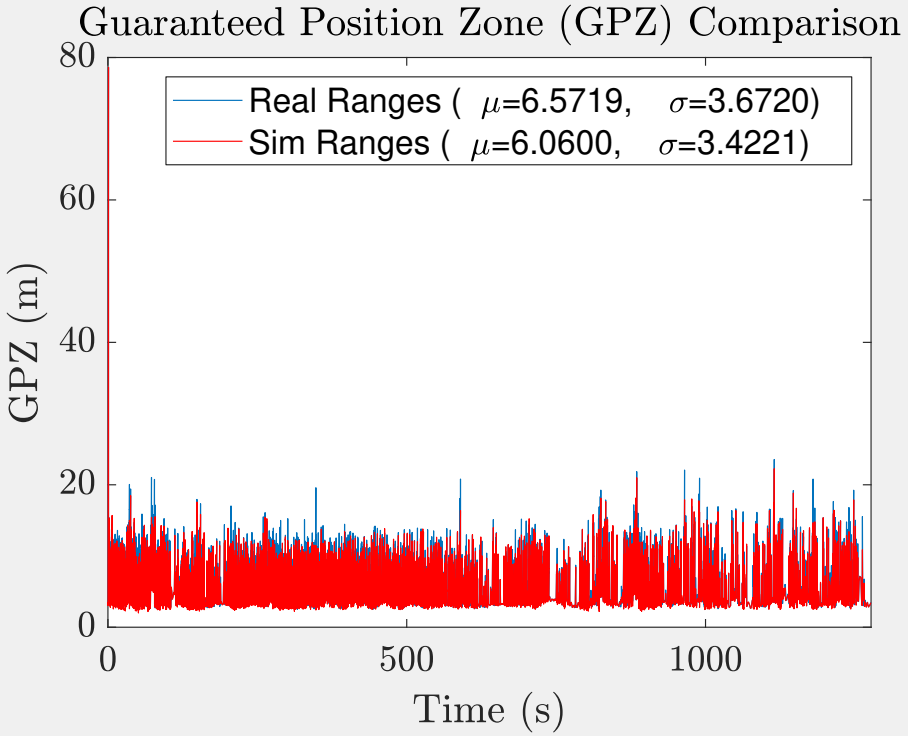

Figure 24: Comparison of the GPZ for Test Scenario 2 using simulated ranges vs. real radio ranges

4.2.4.4 SOM Results

Although the simulated ranges provide a minute advantage over the real ranges in the covariance values due to a lack of outliers and lower covariance, it does not noticeably improve observability. Observability depends on the number of available sensors, the frequency of measurements coming from these sensors, and the geometry. For this reason, it makes sense that there is little difference between the test run with simulated ranges vs real ranges. Both have the same number of ranging radio sensors and both have data measurements at the same timestamps.

In the end, this comparison between the simulated and real ranges reveal that the presence of additional noise and outliers in the real data only minimally impacts the overall results of ARMAS. This is good news, as it implies that these ranging radio sensors are providing reliable measurements.

4.2.5 Test Scenario 3: L1 Progressive Single-Sensor Jamming

The third test scenario consists of the following sensors for the indicated time periods below.

1. 8 L1 Satellites (Begin Jamming at 200s, Jam additional satellite every 100s)
2. 1 Velocity Sensor (0-1277s)
3. 3 Ranging Radio Sensors (Collaborative only: add as needed to maintain integrity)

In this test the collaborative solution pulls in additional ranging radio sensors when observability warnings are detected by SOM. The proprioceptive solution does not pull in ranging radio sensors at all and will only use the L1 and velocity sensors for the whole duration. Unlike Test Scenarios 1 and 2, this test scenario involves L1 satellite jamming. For the ease of comparison, the results over the entire duration of the flight will be compared, followed by zooming in to specific time regions to display the effectiveness of collaboration when certain numbers of satellites are available. Various events are indicated in the plots by vertical dotted lines. A green line indicates a time when a ranging radio is added for collaboration, a red line indicates a time when an L1 satellite is removed, and a magenta line indicates a time when a fault is added to a satellite.

4.2.5.1 ENU Errors: TS3 without Position Sensor

The collaborative instance of ARMAS-SOM performs significantly better than the proprioceptive instance, as is evident through comparing the mean and standard deviation values in Figure 25. The average ENU errors are quite low for the collaborative instance, with the highest being 4.0191 for the north direction. The proprioceptive average ENU errors are as high as -130.1655 for east and as low as -39.5625 for up.

It is on the transition from 4 satellites to 3 satellites where the collaborative instance of ARMAS begins to clearly improve the position error. This makes sense, as in order to have a constrained Global Positioning System (GPS) location, a minimum of 4 satellites is required.

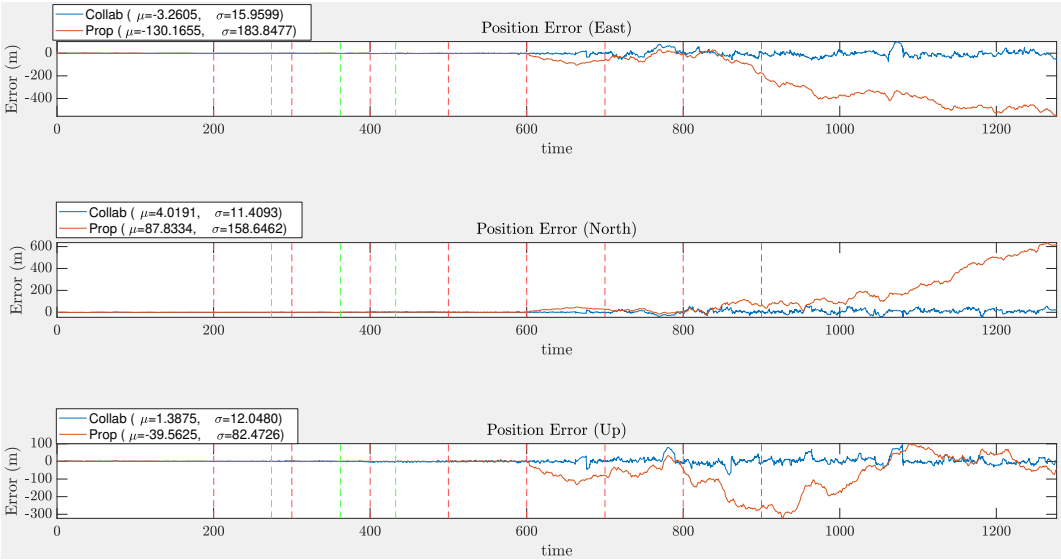

Figure 25: Comparison of the ENU errors for Test Scenario 3.

When there are zero satellites, the ENU positions begin to drift indefinitely for the proprioceptive instance of ARMAS due to the fact that only a velocity sensor is available, which provides relative position only. While observability is lost at this point, it is helpful in demonstrating the effect of collaboration on constraining the position in the presence of no L1 satellites.

4.2.5.2 RSS Error: TS3 without Position Sensor

We can observe from Figure 26 that the collaborative instance begins to improve dramatically compared to the proprioceptive instance on the RSS at $t = 600$s, or when there are only 3 satellites providing information. There are not enough satellites available to obtain a constrained position from the L1 pseudoranges and so these results are as expected. We remind the reader that the observability (e.g. SOM) for the proprioceptive solution becomes unstable at 5 L1 satellites.

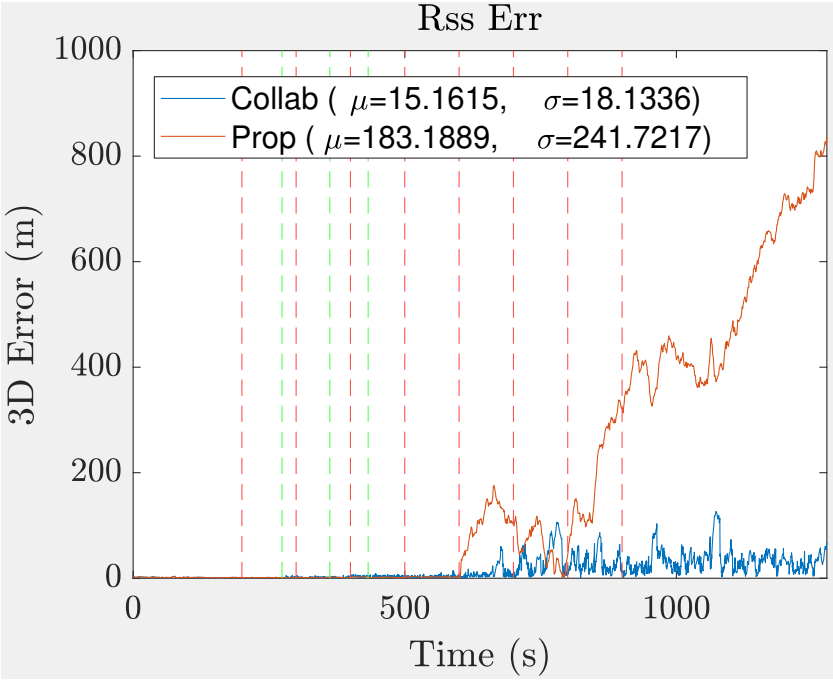

Figure 26: Comparison of the 3D RSS errors for Test Scenario 3

Now we examine in more detail the effect which removing additional satellites has on the RSS error, each region containing 3 (600-700s), 2 (700-800s), 1 (800-900s), and 0 (900-1277s) satellites will be considered. The first region, where there are 3 L1 satellites available, is depicted in more detail in Figure 27. In this region, the mean RSS error is 8.8967 and 105.7241 for the collaborative and proprioceptive solutions, respectively.

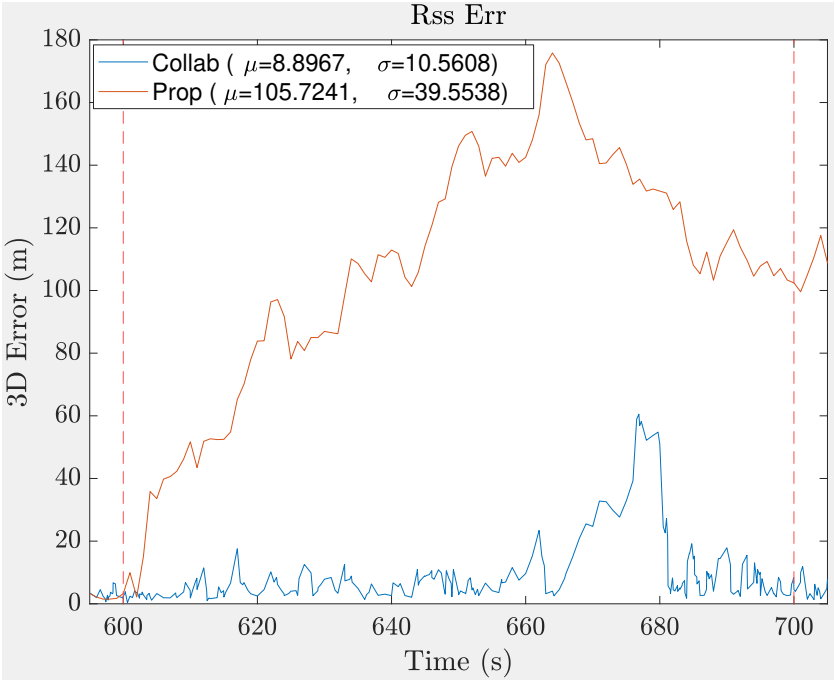

Figure 27: Comparison of the 3D RSS errors for Test Scenario 3 from $t = 600$s to $t = 700$s, consisting of 3 L1 satellites

The next region, consisting of only two L1 satellites in Figure 28, actually produces a better proprioceptive solution than the region consisting of 3 L1 satellites, but by random happen-stance. This is likely attributed to the quality of the noisy velocity sensor data at this period. It could also be attributed to the internal workings of ARMAS. This region still sees an improvement going from the proprioceptive to the collaborative solution, but not as much as the previous region.

It is important to recall that using real ranging radios introduces areas in which there are are little-to-no-ranges for some or all of the radios. This particular region from 700-800s produces 99, 16, and 34 total ranges for radios 101, 102, and 103 respectively (grand total of 149). This is much lower than than that of the region from 800-900s, shown in Figure 29, which produces 17, 133, and 166 ranges for these same radios (grand total of 316). Additionally, there are no ranges from any radio from 736.672s to 750.652s and 764.751s to 789.92s. These are periods of 13.98s and

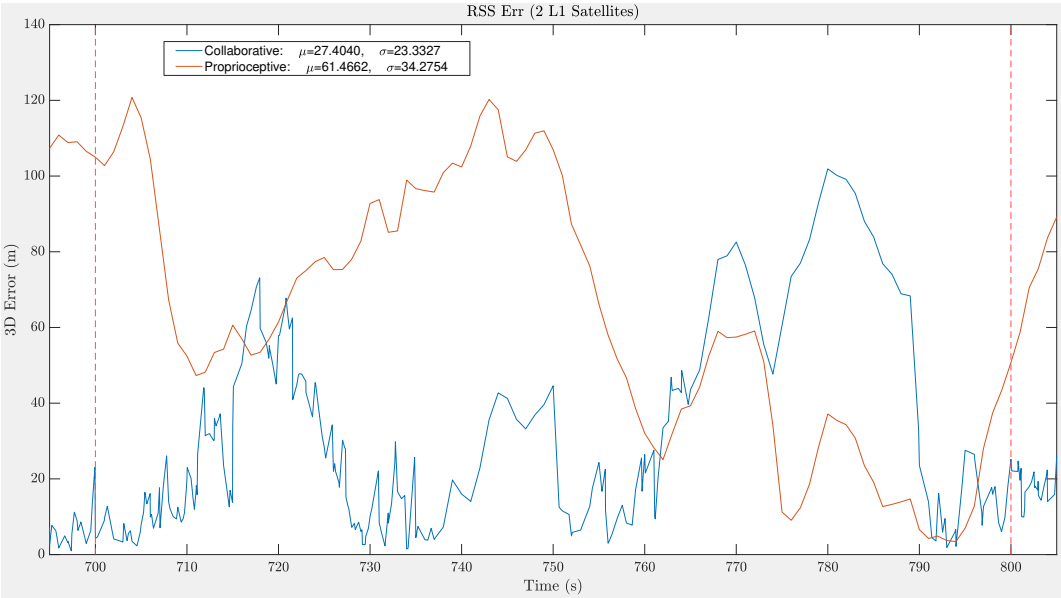

Figure 28: Comparison of the 3D RSS errors for Test Scenario 3 from $t = 700$s to $t = 800$s, consisting of 2 L1 satellites

25.17s without any radio ranges, which explains the poorer performance during these time periods for the collaborative instance of ARMAS.

It not very surprising that that the region shown in Figure 29, consisting of 1 L1 satellite, performs better than the previous region with 2 L1 satellites. This is because the region with 2 satellites experiences multiple ranging radio outages while the region with 1 L1 satellite generally has good ranging radio data (especially for radios 2 and 3). This verifies intuition that one would expect better performance when more radio ranges are available, especially when not very many satellites are available. This region shows a significant improvement in the collaborative instance over that of the proprioceptive instance, with collaboration improving the mean from 196.2712 to 25.6383 and the standard deviation from 89.9861 to 15.8487 when compared to the drifting and unobservable, proprioceptive solution.

In this final region, we see the effects of an unobservable system which relies solely on a relative positioning system, in this case the velocity sensor. The RSS error takes

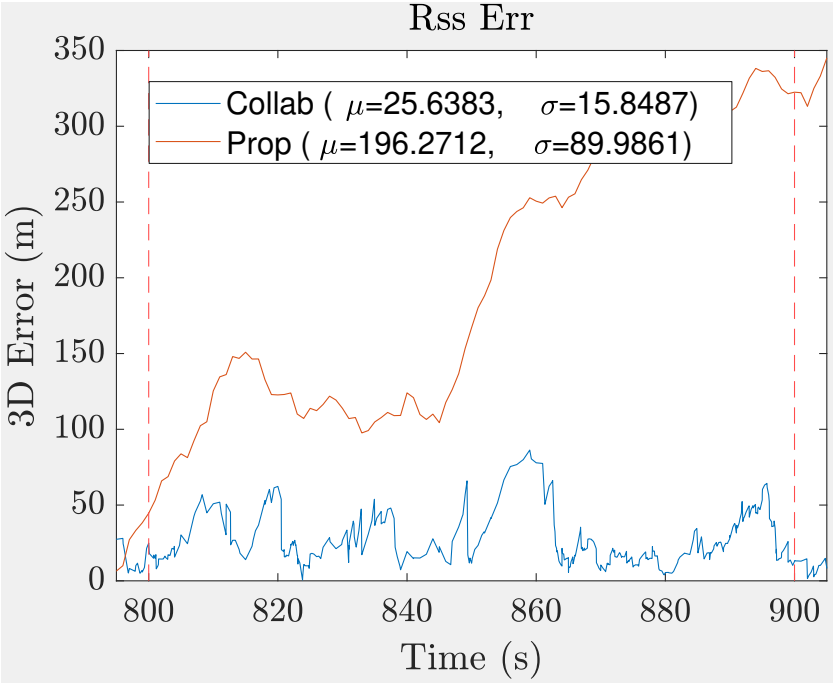

Figure 29: Comparison of the 3D RSS errors for Test Scenario 3 from $t = 800$s to $t = 900$s, consisting of 1 L1 satellite

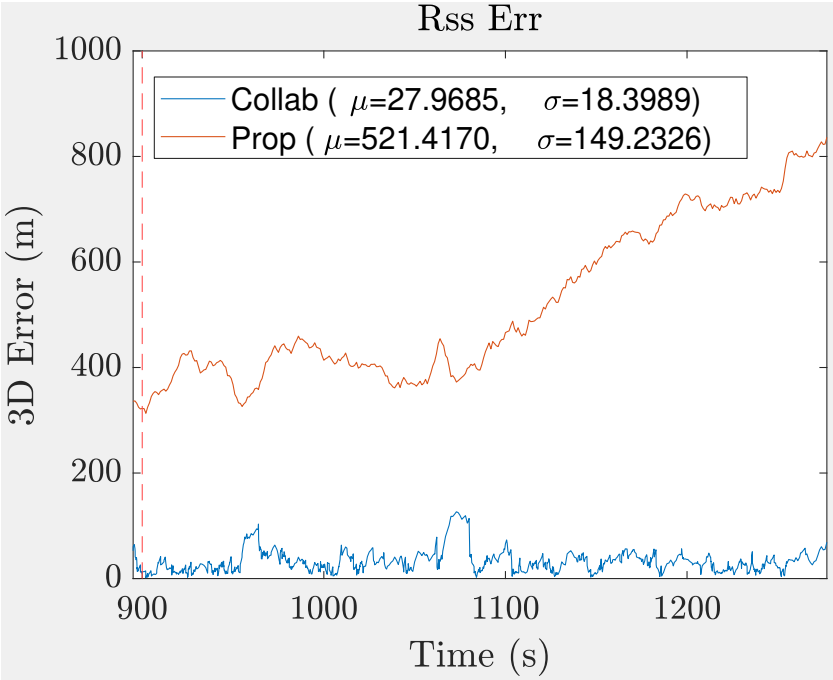

Figure 30: Comparison of the 3D RSS errors for Test Scenario 3 from $t = 900$s to $t = 1277$s, consisting of 0 L1 satellites

off and shows no signs of coming back down in the proprioceptive solution. We also see increases in the average RSS error for the collaborative and proprioceptive solutions than in the previous region with one L1 satellite. For the collaborative instance, the solution is more sensitive to outages in one or more ranging radios due to there being no satellites available to augment these ranging sensors. Either way, this region reveals the effectiveness of collaboration in constraining a system in which only relative position sensors are available.

4.2.5.3 NEES Results: TS3 without Position Sensor

The results above revealed that the errors begin to noticeably improve between the proprioceptive and collaborative solutions at $t = 600$s (3 L1 satellites). Examining the NEES results for this helps give an indication of the estimator's credibility over these same periods.

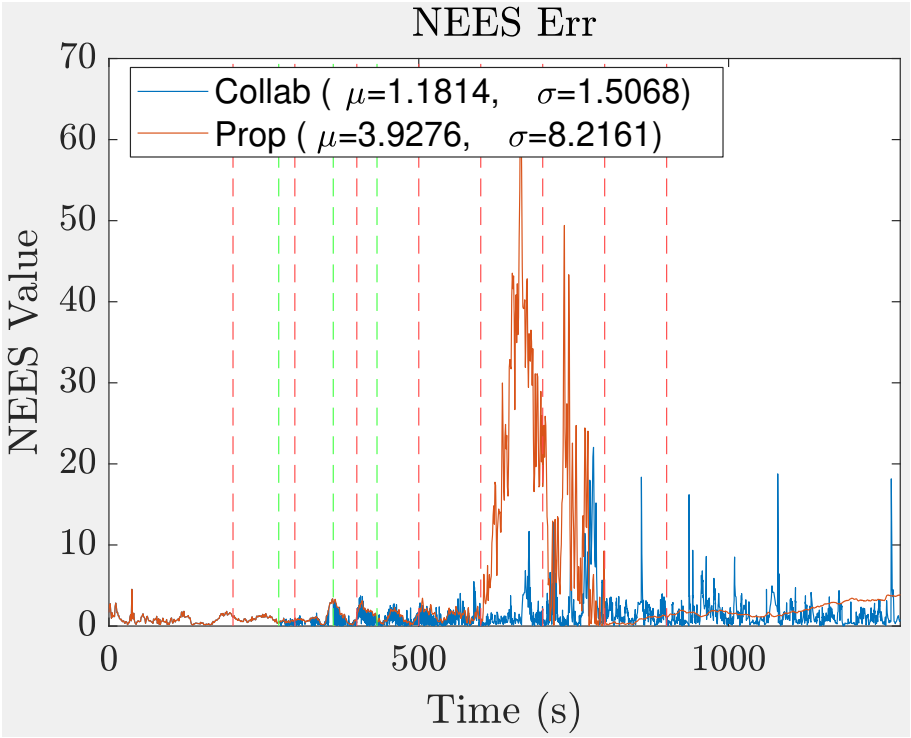

Figure 31: Comparison of the NEES for Test Scenario 3

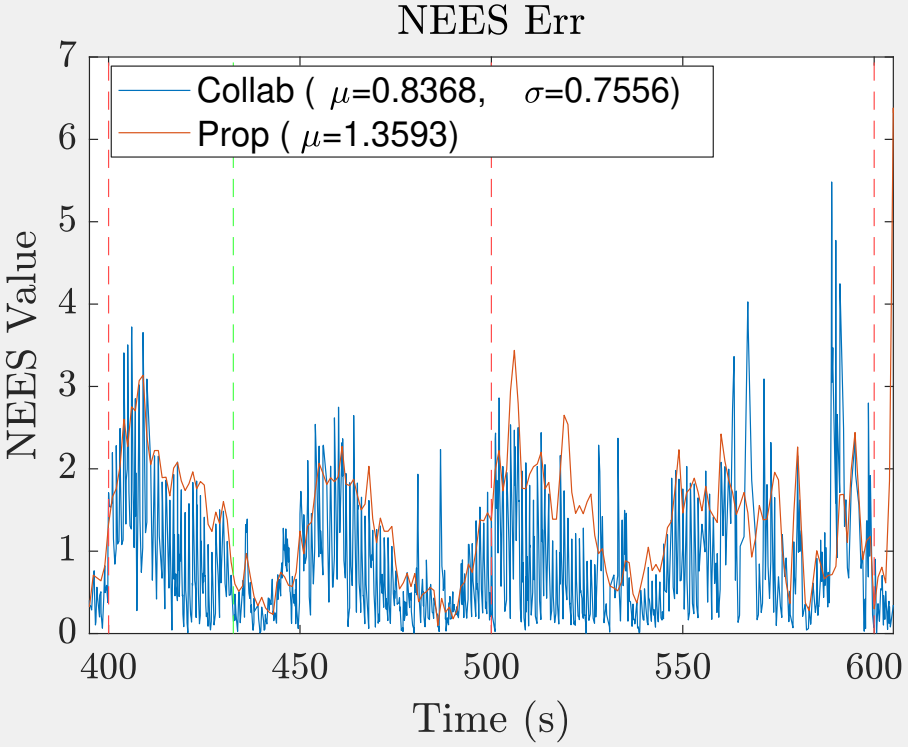

Figure 32: Comparison of the NEES for Test Scenario 3 from $t = 400$s to $t = 600$s, consisting of 4 and 5 L1 satellites

Figure 32 displays the NEES over the time period in which there are 5 L1 satellites $(400-500\text{s})$ and 4 L1 satellites $(500-600\text{s})$. In this region, both average NEES values are not too far from one ($\mu = 0.8368$) for collaborative and ($\mu = 1.3593$) for proprioceptive instances. The collaborative instance also includes the ranging radios, which are bringing the average NEES down. Regardless, both average NEES values still indicate reasonable performance as the proprioceptive and the collaborative instance are not too far away from one. Additionally, although at this point the NEES results are only somewhat better for the collaborative instance, the collaborative instance still maintains observability while the proprioceptive instance does not.

Moving to the region of 3 L1 satellites $(600-700\text{s})$ and 2 L1 satellites $(700-800\text{s})$ it is clear that here is where the average NEES values begin to depart. Although the value of 1.6415 for the collaborative solution is not perfect, it vastly outperforms

the average of the proprioceptive ($\mu = 17.4558$). It is also important to note two observations from here: a) the clear improvement in the results when shifting from 4 to 3 satellites is reinforced as was noted when looking at the RSS error, and b) the region from $700-800$s magnifies the NEES average over this region due to the lack of available radio ranges. In the region of 5 and 4 satellites, the collaborative instance only improves observability of ARMAS and thereby the FDE capabilities over the proprioceptive instance. It is at 3 satellites where all the other values (i.e. position errors, average NEES, and GPZ) in the collaborative instance also vastly outperform the proprioceptive instance.

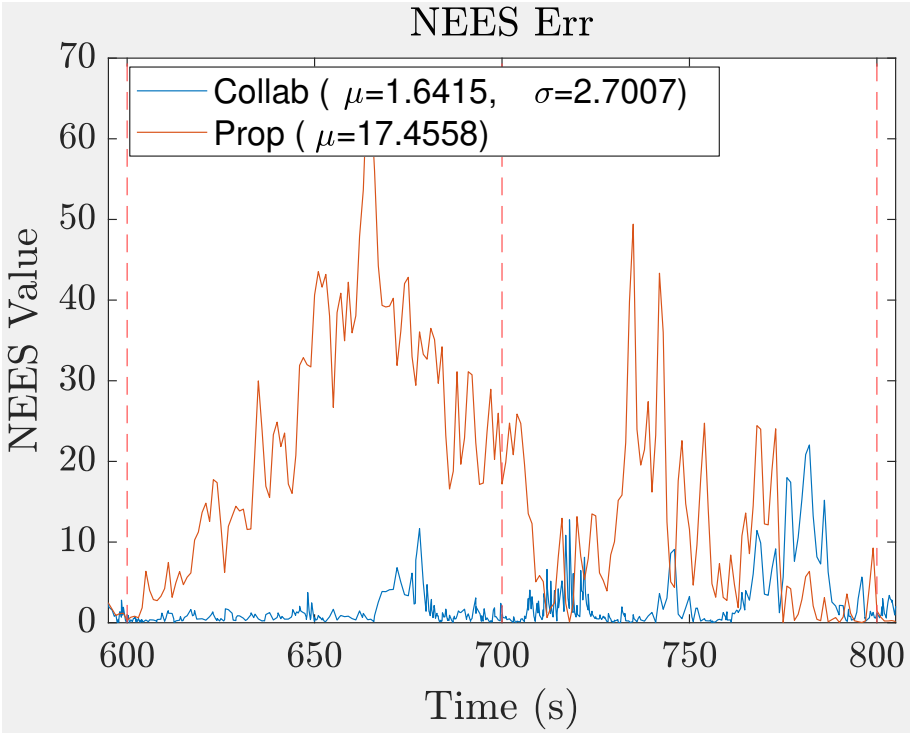

Figure 33: Comparison of the NEES for Test Scenario 3 from $t = 600$s to $t = 800$s, consisting of 2 and 3 L1 satellites

The shift from two to one L1 satellites is where we begin to see the NEES peformance in the proprioceptive instance come back down. The reasons for this will not be extensively considered here, but at this point it is important to remember that

we would not necessarily expect the NEES to get bad at a low number of satellites (unlike RSS and GPZ). For this reason, it is not particularly concerning that the NEES goes back down.

It is safe to assume that the NEES metric is not very useful at this point (less than four L1 satellites). Since the RSS error and GPZ are performing quite poorly and observability is lost, no matter how well the average NEES is, the overall performance will still be very poor. This demonstrates the need for analyzing multiple measurement metrics. While it may seem by looking at just the average NEES that the filter performance is good, it is when you look at the RSS, GPZ, and observability that you recognize the proprioceptive solution is not performing well.

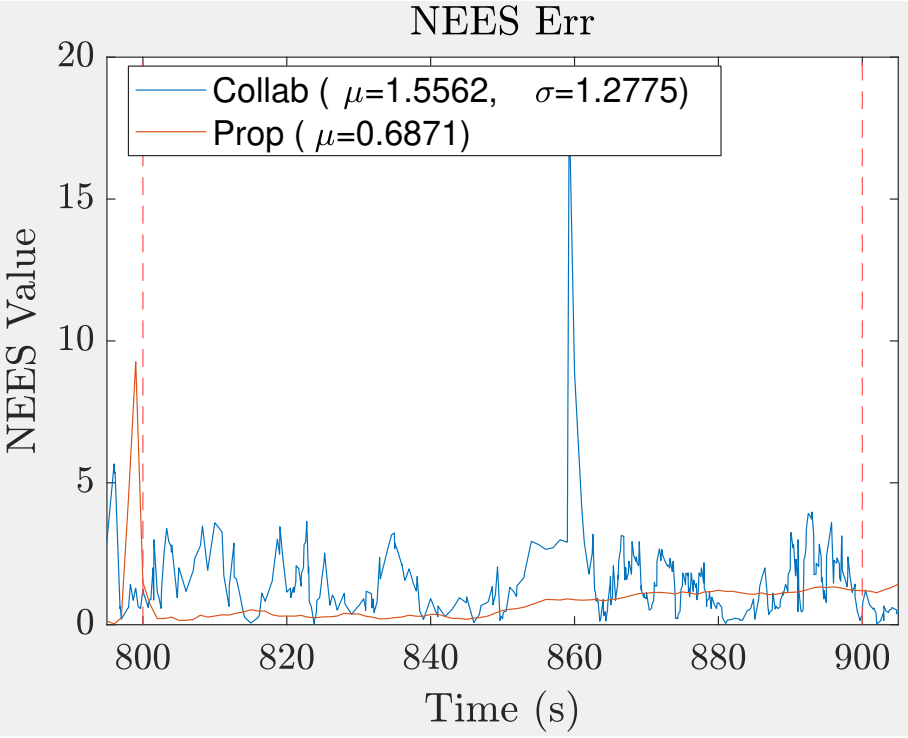

Figure 34: Comparison of the NEES for Test Scenario 3 from $t = 800$s to $t = 900$s, consisting of one L1 satellite

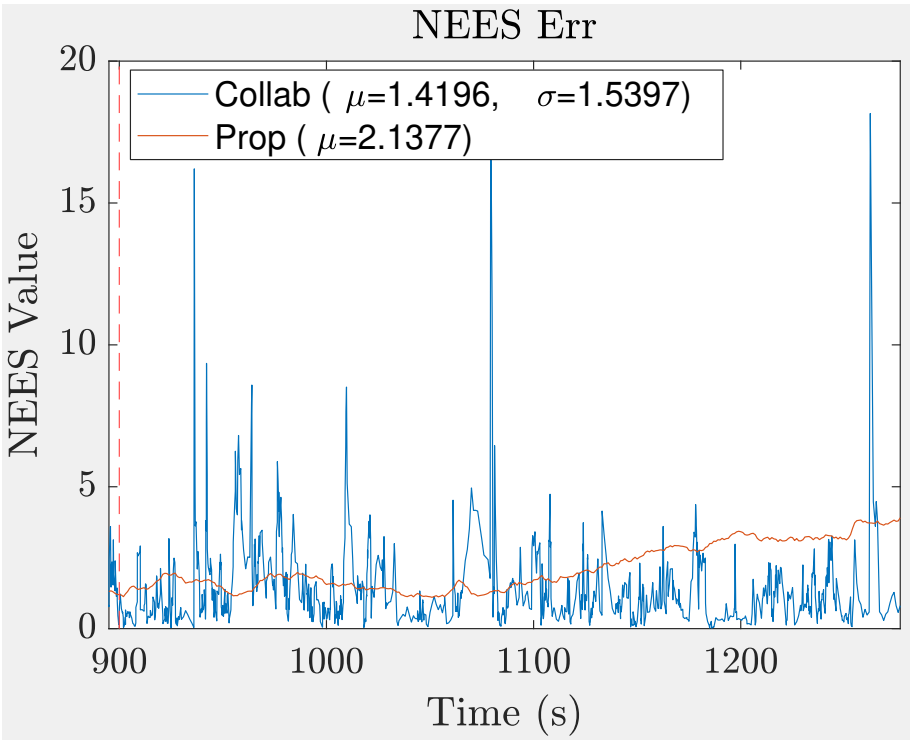

Figure 35: Comparison of the NEES for Test Scenario 3 from $t = 900$s to $t = 1277$s, consisting of 0 L1 satellites

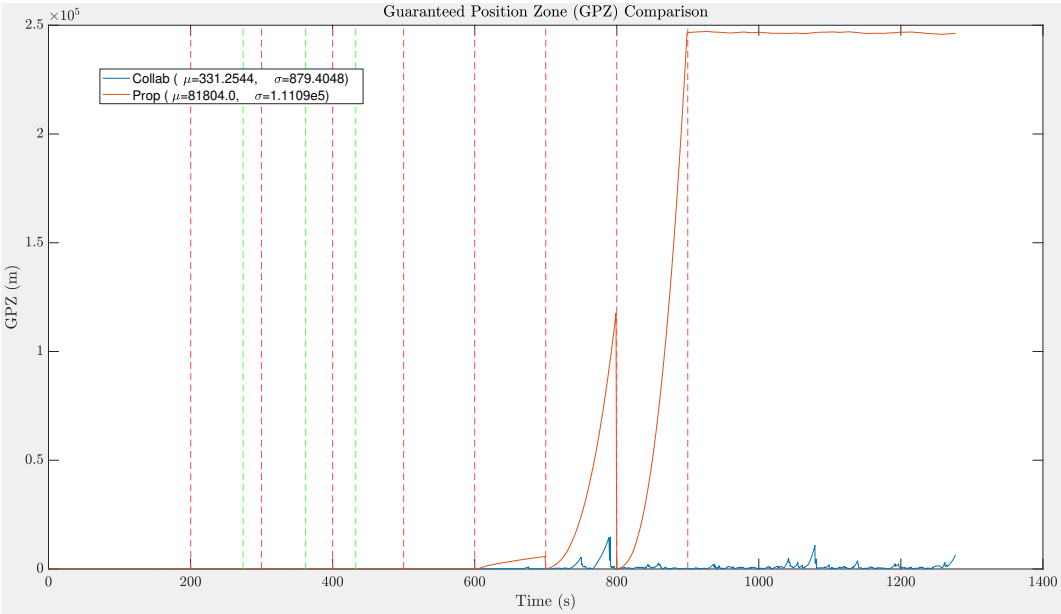

Figure 36: Comparison of the GPZ for Test Scenario 3

4.2.5.4 GPZ Results: TS3 without Position Sensor

The final measurement metric to closely examine here is the GPZ. It is difficult to use to draw very many conclusions from Figure 36 due to the vast change in scale of the GPZ. For this reason, as with the RSS error and NEES values, various regions are focused on in more detail.

It is on the shift from 6 to 5 satellites where we begin to see a consistent improvement on the GPZ from the collaborative solution over the proprioceptive solution (see Figure 37. Figure 38 shows this improvement is even more noticeable when shifting from 5 to 4 satellites ($t = 400 - 600s$). Due to the nature of the GPZ being the union of the FDE filtering layer covariances, the removal of any sensor is expected to make the GPZ worse.

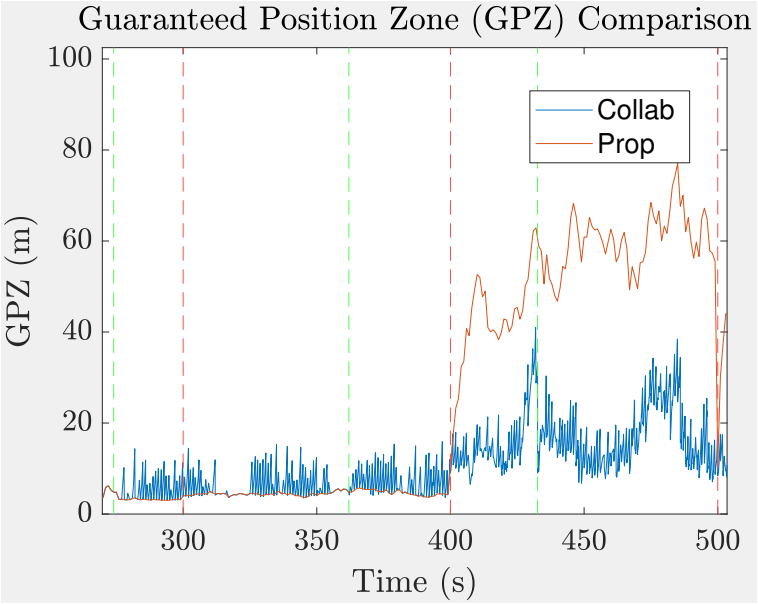

Figure 37: Comparison of the GPZ for Test Scenario 3 from $t = 350$s to $t = 500$s, consisting of 6 and 5 L1 satellites

The RSS error and NEES results revealed significant downgrades in the proprioceptive performance on the shift from 4 to 3 L1 satellites. For this reason, it is unsurprising that the GPZ also experiences this as seen in Figure 39. When shifting

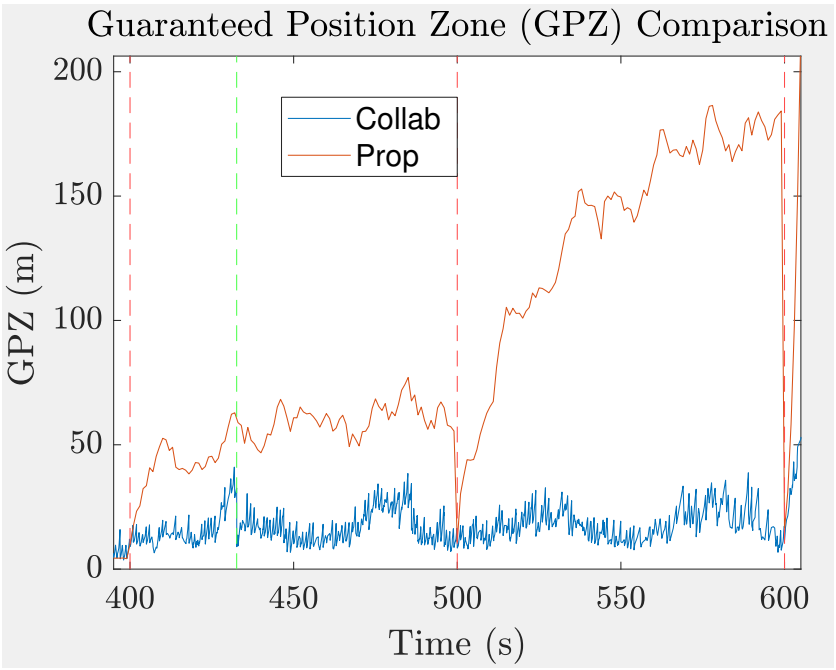

Figure 38: Comparison of the GPZ for Test Scenario 3 from $t = 450$s to $t = 600$s, consisting of 5 and 4 L1 satellites

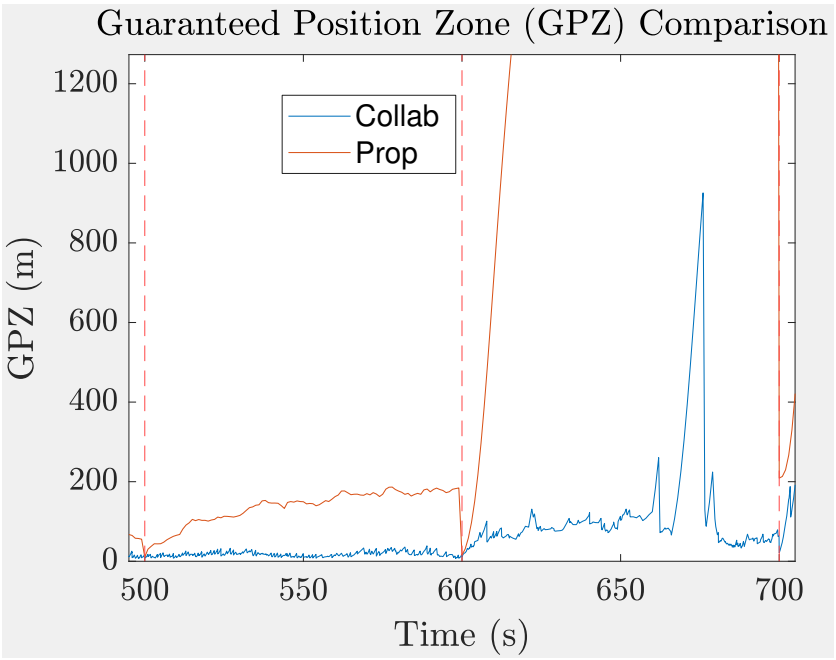

Figure 39: Comparison of the GPZ for Test Scenario 3 from $t = 550$s to $t = 700$s, consisting of 4 and 3 L1 satellites

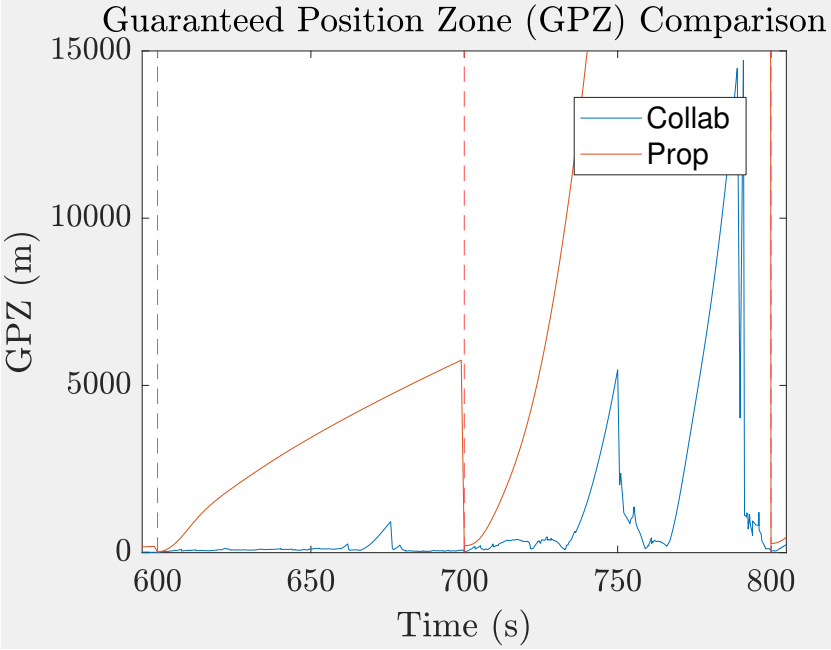

Figure 40: Comparison of the GPZ for Test Scenario 3 from $t = 650$s to $t = 800$s, consisting of 3 and 2 L1 satellites

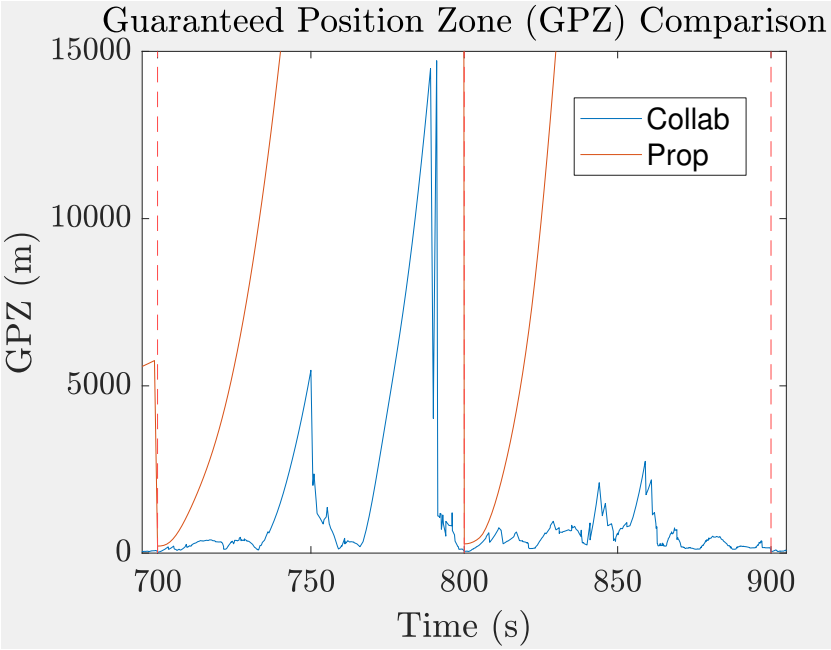

Figure 41: Comparison of the GPZ for Test Scenario 3 from $t = 750$s to $t = 900$s, consisting of 2 and 1 L1 satellites

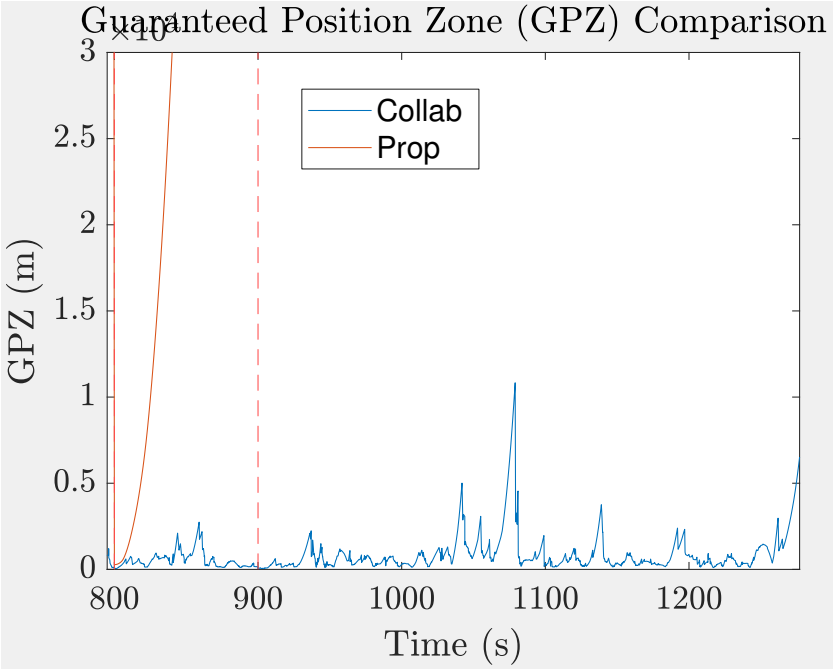

Figure 42: Comparison of the GPZ for Test Scenario 3 from $t = 850$s to $t = 1277$s, consisting of 1 and 0 L1 satellites

from 4 to 3 L1 satellites, it becomes difficult to have the collaborative and proprioceptive instances on the same plot. Additionally, the effect of sporadic ranges between radios is shown to dramatically impact the GPZ for three or less satellites.

The removal of additional satellites reveals: a) the GPZ dramatically increases when removing satellites, especially under 4 satellites, and b) the periods of no ranges exponentially increases the GPZ when less satellites are present.

The plot from Figure 41 reiterates previous points and shows how drastically the ranging radio outage from $764 - 789$s impacts the GPZ. This outage, and a generally poor ranging radio performance in comparison to other regions of the flight, reveal the need for consistent ranging radio conversations.

This final region shows that with just the 3 ranging radios accompanying the velocity sensor, the GPZ is not great and is prone to much error. Nonetheless, collaboration clearly and consistently improves the GPZ.

4.2.5.5 SOM Results: TS3 without Position Sensor

It is in test scenario 3 where the SOM results begin to become inconsistent when tested with real data. This test scenario is intended to extensively test the capabilities of SOM in detecting observability threats and adding sensors when needed to help maintain observability in the midst of L1 satellite jamming. The good news is that SOM added the three ranging radios when observability warnings were detected. While the logic used to add additional sensors needs to be tweaked compared to that of previous tests with the real data (because of the temporary observability warnings mentioned above causing premature collaboration), SOM was still able to detect observability warnings and add ranging radios. The advantage of collaboration is also clearly shown in the RSS, NEES, and GPZ results.

Even though there was a clear success in the employment of SOM in that the framework successfully added sensors for collaboration, there were also some concerning results. These results were seen when there were less than 2 L1 satellites available for the collaborative solution. The proprioceptive solution produces more consistent results in this respect. For example, once there are 5 L1 satellites or less, the proprioceptive solution remains unstable or lost as more satellites are removed, with the exception of occasional time indices with stable observability reported. This is not completely consistent and should be further examined, but overall, once the proprioceptive solution loses observability, we would expect it to continue in this as more satellites are removed.

The collaborative results are more perplexing and must be studied extensively. In one sense, the collaborative results are expected to be less consistent due to the use of 3 ranging radio sensors which produce intermittent measurements, with outages up to 25 seconds. Nonetheless, we would not expect the collaborative results to go from fluctuating between unstable and stable with 3 and 2 L1 satellites to consistently

stable with 1 and 0 L1 satellites. In this case it seems that SOM is producing erroneous results at 0 and 1 L1 satellites. We can see, from test scenario 5 (section 4.2.7), that the collaborative instance is no longer able to detect a bias when there are 2 L1 satellites left. This indicates that observability is not present from 2 satellites and less, yet SOM is indicating stability with 0 and 1 L1 satellites. In light of this apparent contradiction, these results must be looked into further.

4.2.6 Test Scenario 4: Jam 5 L1 Sensors Simultaneously

The fourth test scenario consists of the following sensors for the indicated time periods below:

1. 8 L1 Satellites (Jam 5 L1 Sensors at $t = 600$s)
2. 1 Velocity Sensor (0-1277s)
3. 1 Position Sensor (0-1277s)
4. 3 Ranging Radio Sensors (Add as needed to maintain integrity)

Removing multiple L1 sensors at a time reveals whether or not ARMAS responds the same when jamming multiple satellites as opposed to jamming satellites one-by-one. In this test scenario, 5 L1 sensors are removed at $t = 600$s. This is the same time where the sixth satellite is removed in Test Scenario 3, with the only difference being that the satellites are removed at the same time. We would expect nearly identical results at this point ($t = 600$s), due to the same sensors being available at this same time. However, we will see in the following results that this is not necessarily so.

4.2.6.1 RSS Error: Test Scenario 4

Figures 43 and 44 reveal a noticeable improvement in the overall RSS error in the collaborative solution over the proprioceptive ($\mu = 5.5601$ vs. $\mu = 11.5932$). The

discrepancy between the collaborative and proprioceptive solutions does not occur until almost 700s.

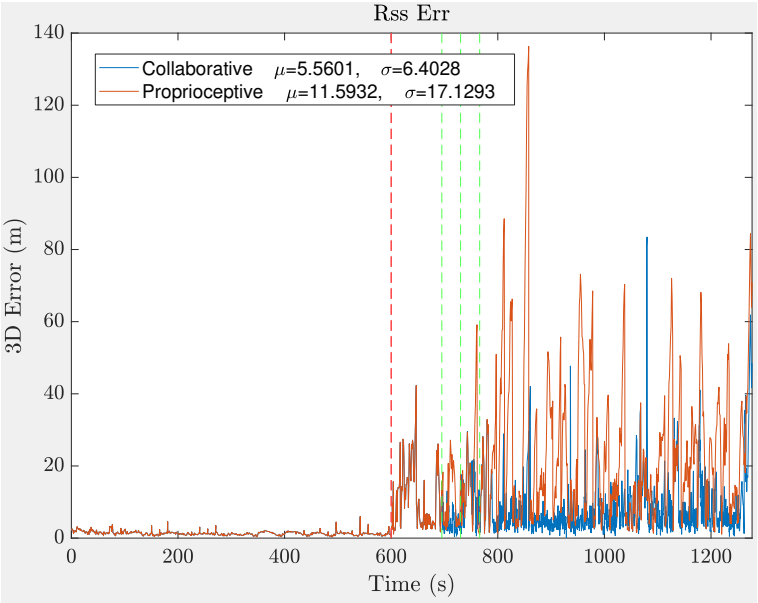

Figure 43: Comparison of the RSS for Test Scenario 4 from $t = 0$s to $t = 1277$s

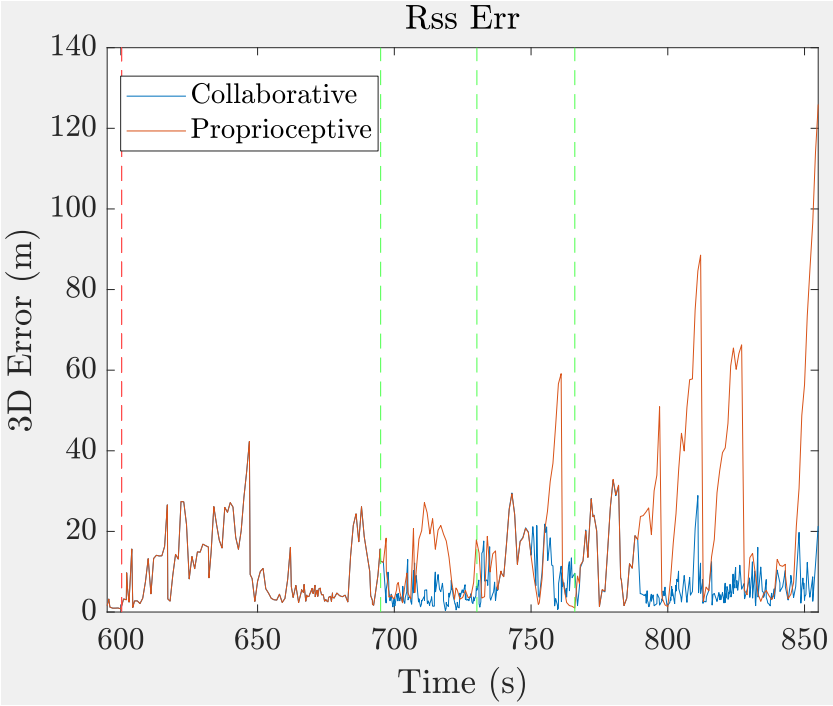

Figure 44: Comparison of the RSS for Test Scenario 4 from $t = 600$s to $t = 850$s

Figure 44 helps clearly show what happens at the switch from eight L1 satellites to three L1 satellites at $t = 600$s. It takes ARMAS almost 100s after the satellite jamming event for the first L1 satellite sensor to be removed. The third and final ranging radio is not added until 176s after the jamming event. This reveals again the inconsistency of SOM when tested with real data. Although SOM was unstable at 3 satellites in TS3, when these satellites are removed at simultaneously, SOM does not give the same result (even though the available sensors at this point are identical for the proprioceptive solution). In this scenario, SOM only reports temporary observability threats, reiterating the need to examine the inner-workings of SOM when testing real data.

4.2.6.2 NEES: Test Scenario 4

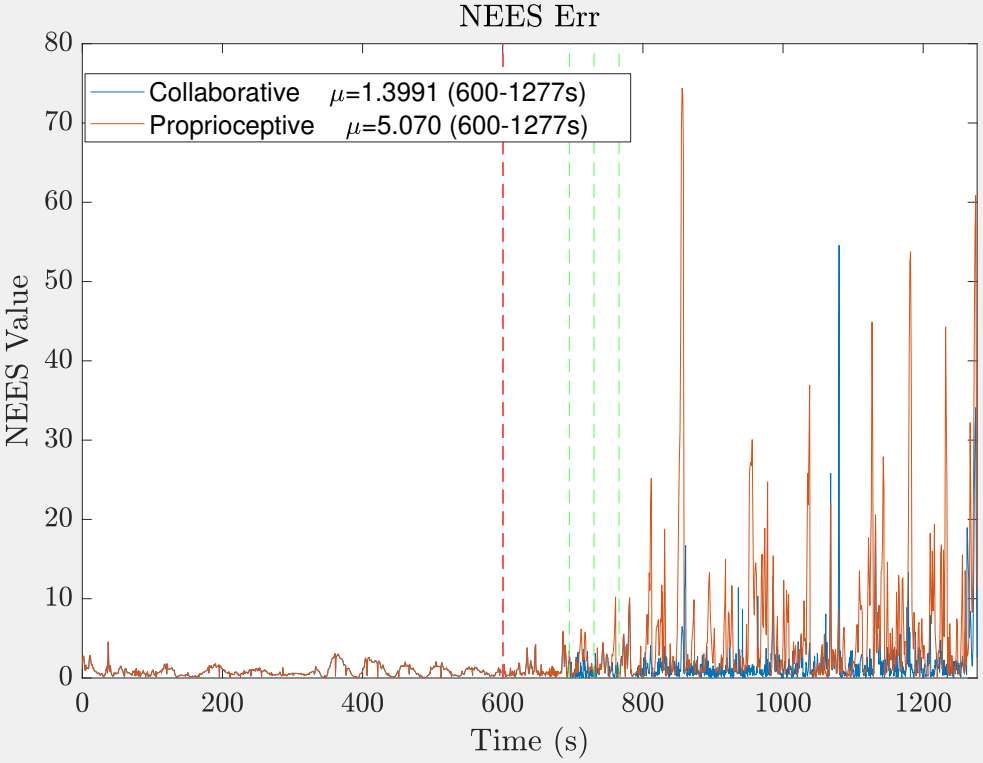

Figure 45: Comparison of the NEES for Test Scenario 4 from $t = 0$s to $t = 1277$s

There is not much to say about the NEES for TS4 other than the collaborative solution produces significantly better results than the proprioceptive. The average NEES of the collaborative solution ($\mu = 1.3991$) is much closer to one than that of the proprioceptive ($\mu = 5.070$) after the sensors are removed.

4.2.6.3 GPZ: Test Scenario 4

As with the NEES and the RSS error, the collaborative GPZ outperforms the proprioceptive GPZ. This is no surprise, as the proprioceptive solution does not have enough satellites for a constrained solution while the collaborative has the ranging radios to compensate for this deficiency.

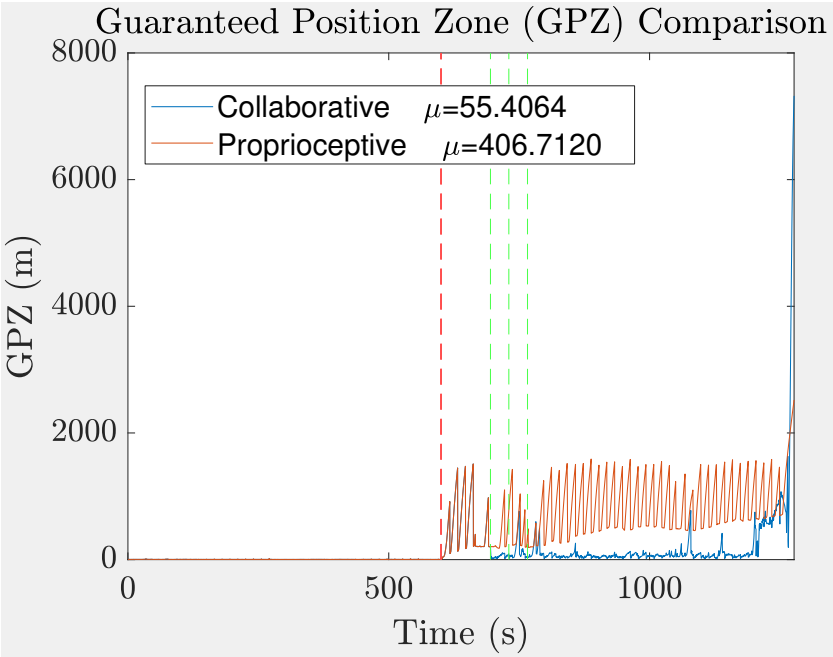

Figure 46: Comparison of the GPZ for Test Scenario 4 from $t = 0$s to $t = 1277$s

Overall, the collaborative instance vastly outperforms the proprioceptive in every measurement metric once the satellites are removed. Nonetheless, more SOM inconsistencies are brought to light. Even though this test scenario consists of the same exact sensors as TS3 at $t = 600$s (with the difference being simultaneous jamming vs.

one-by-one satellite jamming), different SOM results are reported. For this reason, and many others, SOM needs to be tested and examined extensively.

4.2.7 Test Scenario 5: L1 Bias Added

The fifth test scenario tests ARMAS when a sensor experiences a bias (spoofing). This test scenario consists of the following:

1. 8 L1 Satellites (Begin Bias at 200s, Spoof additional sat every 100s)
2. 1 Velocity Sensor (0-1277s)
3. 1 Position Sensor (0-1277s)
4. 3 Ranging Radio Sensors (Collaborative only: add as needed to maintain integrity)

This setup is identical to that of test scenario 3 with the main difference being that every 100s beginning at 200s, an additional L1 satellite has a bias introduced rather than being jammed. This test scenario uniquely tests the effect of collaboration on ARMAS-SOM's FDE capabilities.

4.2.7.1 RSS Error: Test Scenario 5

In Figure 47, two collaborative and two proprioceptive instances of ARMAS are compared for the purpose of demonstrating how the removal of the position sensor impacts this test scenario. First, the collaborative and proprioceptive solutions will be compared when the position sensor is and is not present. This comparison will be followed by analyzing some other important observations about the faults detected in this test scenario.

The 3D RSS error plots in Figures 47 and 48 reveal an improvement in the collaborative instance in the overall mean and standard deviation values. These plots, most

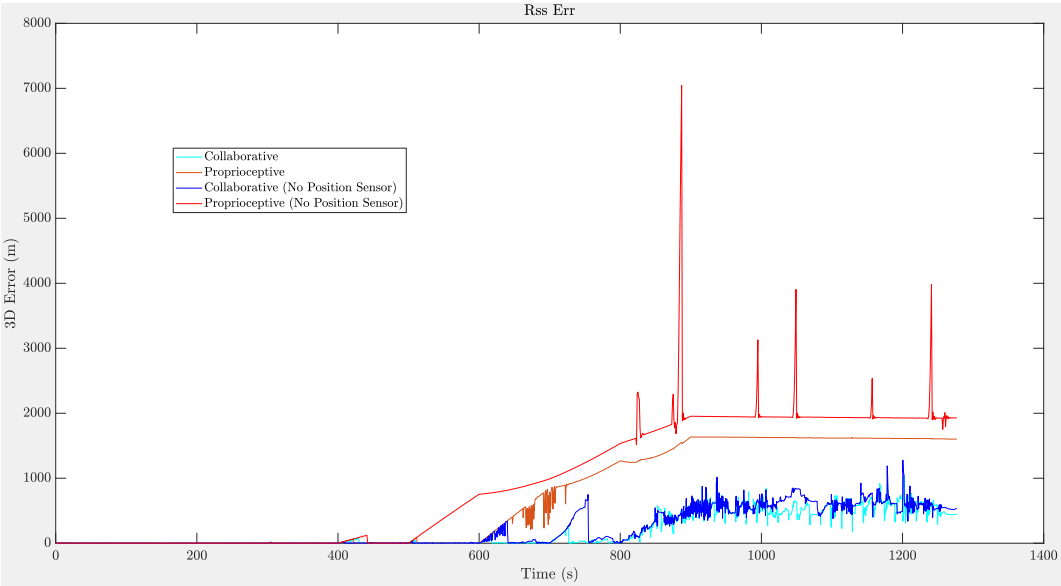

Figure 47: Comparison of the 3D RSS error for Test Scenario 5

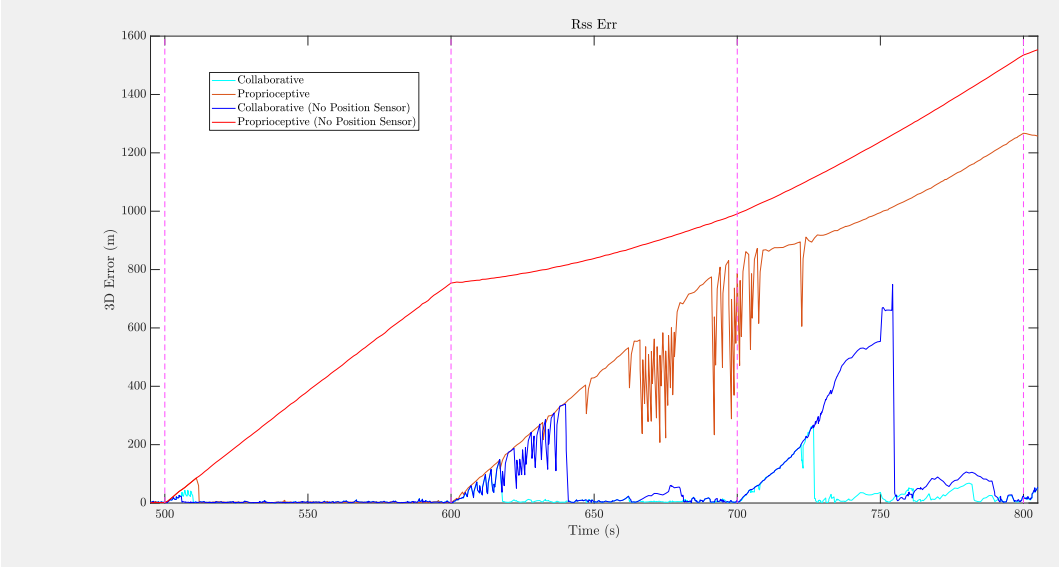

Figure 48: Comparison of the 3D RSS error for Test Scenario 5 during periods of 2,3, and 4 L1 satellites

easily seen in Figure 48, reveal that the addition of three ranging radios allows for three more faults to be detected in the L1 satellites when there is no position sensor present. However, when the position sensor is present, the proprioceptive solution is able to detect one more satellite bias, while the collaborative solution still detects

the same number. The presence of a position sensor decreases the amount of time to detect the bias instituted at 700s for the collaborative solution, nonetheless it is still unable to detect an additional bias at $800s$. In general, these tests show that the addition of a position sensor does improve FDE capabilities with four or less satellites present, yet this advantage is more pronounced in the proprioceptive solution.

Finally, the effects of the ranging outage from 736.627 to 750.652 must be noted. It is no coincidence that at this exact time of 750.652s, which is where the radio outage ends, the faulty L1 satellite is removed in the scenario with no position sensor. For this reason, it is reasonable to assume this fault would have been detected earlier had there been more successful radio ranges available. This also suggests that had there been more ranges been available from $800 - 900$s, then another fault would be detected here for the scenario with the position sensor. Either way, these results confirm the need for consistent sensor measurements.

4.2.7.2 NEES Results: Test Scenario 5

It is from $0-40$ seconds where the NEES performs best ($\mu = 1.5002$ from $0-400$). This mean value is skewed by the temporary vast increases in NEES. Removing three of the highest data points (out of 610 total data points) from $0 - 400$s brings this mean down in the collaborative solution to 1.0463 for this time period. However, once the fourth L1 satellite experiences a bias, which is where there are four total L1 satellites, the average NEES value shoots up due to the bias taking longer to detect (see Figures 49 and 50).

A vast increase in the NEES is seen when the biases are added to the 3rd, 4th, and 5th satellites at the times of 500, 600, and 700 seconds. The NEES, increases significantly until the fault is detected in the satellites. It is also during these times that the effects of collaboration are most clear. The proprioceptive solution is not able

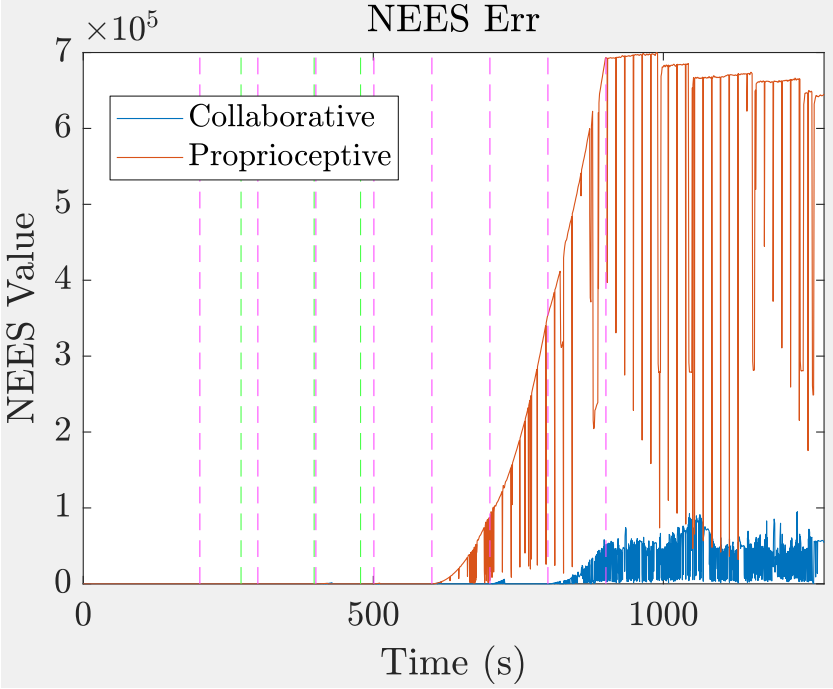

Figure 49: Comparison of the NEES for Test Scenario 5

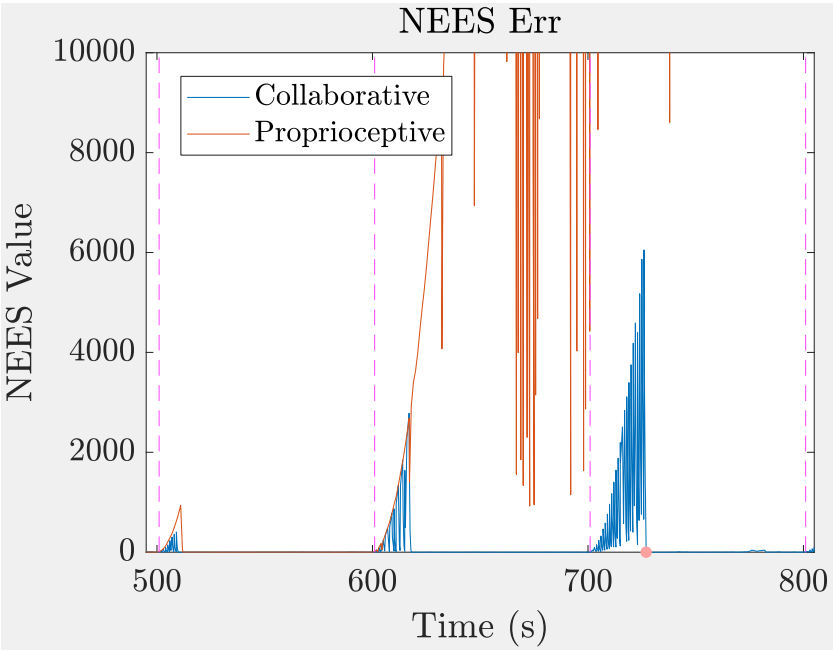

Figure 50: Comparison of the NEES for Test Scenario 5 during periods of 2,3, and 4 L1 satellites

to detect the faults from four L1 satellites and on, while the collaborative instance detects faults until two L1 satellites.

4.2.7.3 GPZ Results: Test Scenario 5

It is when the fifth (fourth when no position sensor is present) satellite experiences a bias where the proprioceptive and collaborative instances depart in their GPZ. It is at this point that the proprioceptive solution loses observability and is unable to detect any more faults. For this reason, its GPZ takes off while the collaborative instance remains constrained. Nonetheless, the collaborative GPZ is affected by the biases to some extent. We see immediate decreases in the GPZ when the faults are detected in these satellites. Similar to previous test scenarios, the magnitude of the GPZ is higher in regions where there are little to no radio ranges.

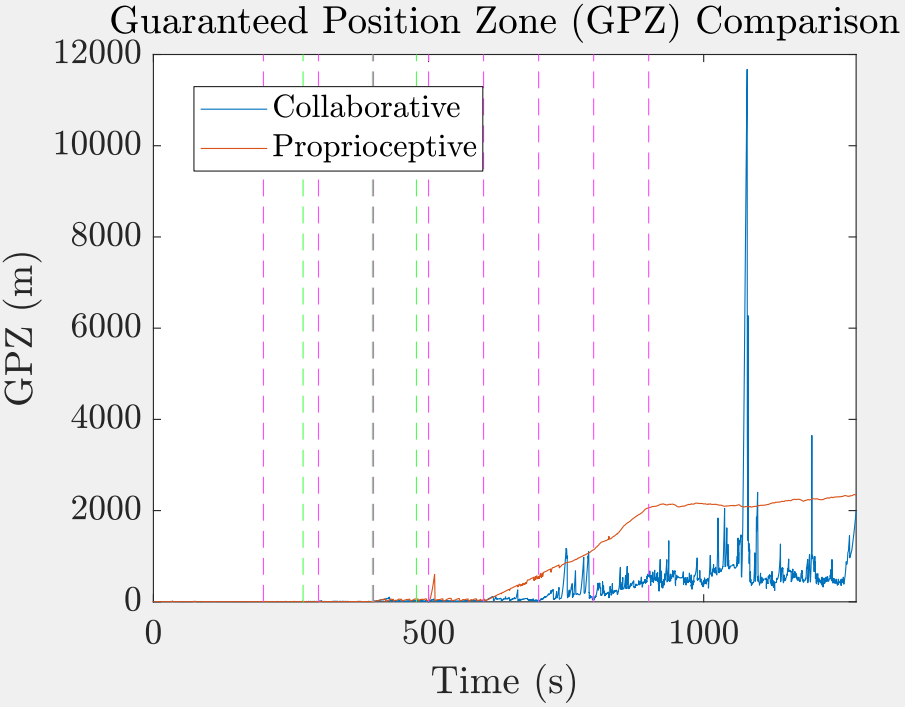

Figure 51: Comparison of the GPZ for Test Scenario 5

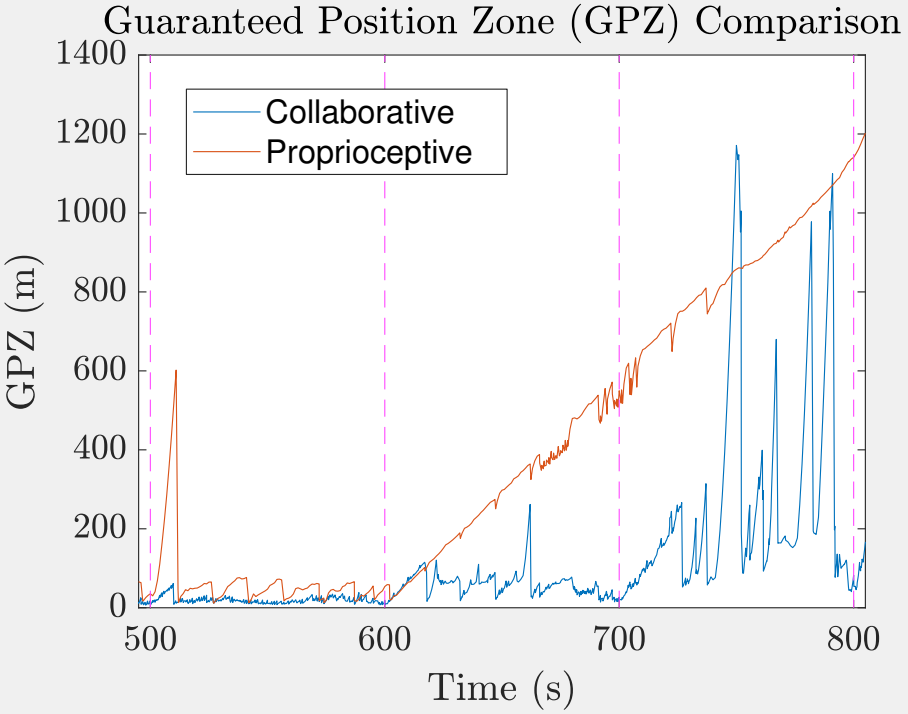

Figure 52: Comparison of the GPZ for Test Scenario 5

4.2.7.4 SOM Results: Test Scenario 5

While the SOM results in TS3 produced some concerning results, the results from TS5 were even more problematic. For example, once the collaborative and proprioceptive solutions are unable to detect a fault (which indicates SOM instability), SOM actually indicates stability with no exceptions. Both the proprioceptive and collaborative solutions indicate stability of SOM throughout the entire duration of the test run (with the exception of temporary instability). The stability of the collaborative solution becomes even more consistent at the point in which faults can no longer be detected. Although, based only on the fault detection capabilities, it is reasonable to expect SOM to become unstable/lost at 4 satellites for the proprioceptive solution and at 2 satellites for the collaborative solution, this is not the case. For this reason, SOM must be further looked into, and the discrepancy between the observability in TS3 and TS5 even when the same number of satellites are available must be accounted

for.

Overall, these results from TS5 indicate the effectiveness of collaboration in maintaining FDE capabilities in the presence of L1 satellite biases. The addition of three radios allows for two (with position sensor) or three (without position sensor) more faults to be detected among the L1 satellites. In spite of this clear advantage of collaboration, that is, an improvement in FDE capabilities, the SOM results are problematic and need to be examined deeply.

4.2.8 Test Scenario 6: L1 Jamming + L1 Bias

The final test scenario tests ARMAS when 4 L1 satellites are jammed and a bias is experienced. This test scenario consists of the following sensors for the indicated time periods:

1. 8 L1 Satellites (Jam 4 Simultaneously + Add Bias to 1 at $t = 500$s)
2. 1 Velocity Sensor (0-1277s)
3. 1 Position Sensor (0-1277s)
4. 3 Ranging Radio Sensors (Collaborative only: 0-1277s)

For this test, the P440 radios were added in at the very beginning for the entire duration. This is done because this test (along with Test Scenario 4) reveals that SOM is unable to detect a threat to observability when multiple L1 sensors are jammed simultaneously. In other words, when four satellites are jammed at once, SOM is unable to detect the threat to observability and thereby add more sensors. In test scenario 3, the satellites were removed one by one every 100s, with the fourth satellite being removed at $t = 500$s. Since both scenarios result in the same amount of sensors at $t = 500$s, we would expect identical observability results. Yet, as this test scenario and test scenario 4 demonstrates, the nature of performing simultaneous jamming

affects SOM differently than jamming sensors one by one over a period of time. The following sections dig deeper into how simultaneous jamming and bias affects the RSS and NEES.

4.2.8.1 RSS Error: Test Scenario 6

The RSS error plots in Figures 55-54 immediately reveals that the collaborative instance is able to detect the sensor fault within 32 seconds while the proprioceptive solution is unable to detect the fault at all. This results in a constant increase in the overall RSS error in the proprioceptive solution due to the increase in satellite bias while the RSS error for the collaborative instance remains low.

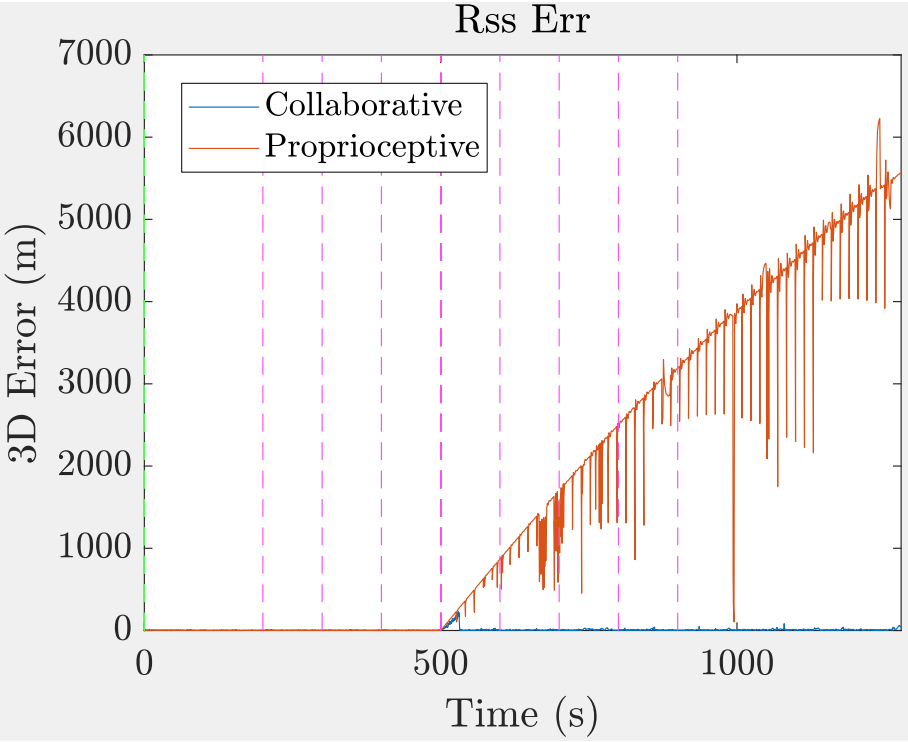

Figure 53: Comparison of the RSS for Test Scenario 6

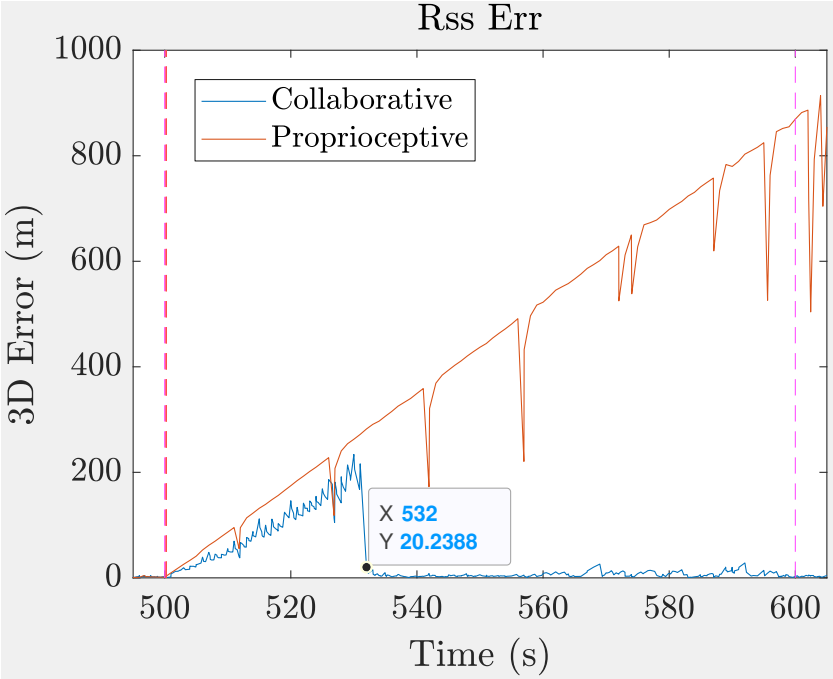

Figure 54: Comparison of the RSS for Test Scenario 6 from $400 - 500$s

4.2.8.2 NEES Results: Test Scenario 6

The NEES results confirm the results from the RSS error. From the NEES results we see a vast discrepancy between the NEES when the fault is present but not detected and all the other times. For example, the average NEES is 0.5637 from $0-500$s, which is prior to the jamming of the satellites and the bias. From $500 - 532$ seconds, the average NEES is 1184.1 and from $532 - 1277$s the average NEES is 1.3074. These result make intuitive sense, as we would expect the NEES to be poor while a fault is present. Additionally, we would expect the NEES to be nearer to one when there are no faults present.

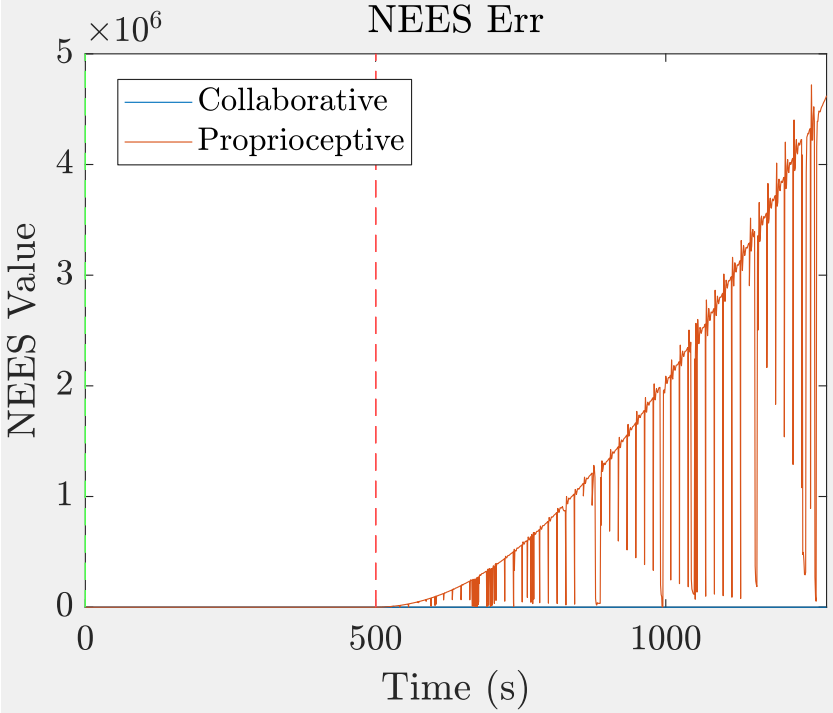

Figure 55: Comparison of the NEES for Test Scenario 6

4.3 Results and Analysis Summary

Overall, there were some positive and negative results discovered through testing the ARMAS-SOM framework with real data. The biggest positive includes successful integration of ARMAS-SOM with real, asynchronous data. While there were some resulting issues which cannot be ignored and must be looked into (i.e. SOM oddities with low number of L1 satellites, discrepancies in SOM between TS3 and TS5, etc.), the effectiveness of collaboration with ranging radios was demonstrated successfully. In other words, the most notable successes have to do with the ability of ARMAS to add collaborative information when observability is threatened, and the demonstration of the effectiveness of this collaborative information. The biggest concerns were revealed in SOM's volatility in areas with borderline observability and SOM's false assurance of stability when it is clear that stability is not possible.

Table 10 presents a summary of the average values for each of the measurement

metrics in each test scenario. From this table, it is clear that the collaborative instance performs better overall. Six out of seven comparisons reveal better results for the collaborative instances of ARMAS. The only instance in which the proprioceptive instance performs the same is when the L1 satellites are available for the entire duration of the test. This indicates that performing collaboration when observability is not threatened provides no noticeable advantage, although unnecessarily performing collaboration does not degrade performance. Take note that the high values in TS5 and TS6 in this table are due to the time periods when a bias is present and not detected.

SCENARIO	Mean RSS Error (m)	ANEES	Avg GPZ
TS1 (Prop)	1.3705	0.9331	3.8204
TS2 (Collab)	1.3825	0.9704	3.7448
TS3 (Prop with Vis)	18.6801	1.3103	1.43E+03
TS3 (Collab with Vis)	12.3001	1.1651	155.4932
TS3 (Prop No Vis)	183.1889	3.9276	8.18E+04
TS3 (Collab No Vis)	15.1615	1.1814	197.1984
TS4 (Prop)	11.5932	5.07	406.712
TS4 (Collab)	5.5601	1.3991	55.4064
TS5 (Prop with Vis)*	692.4233	2.29E+05	855.3009
TS5 (Collab with Vis)*	224.7401	2.57E+04	166.4592
TS5 (Prop No Vis)	928.8741	5.06E+05	4.53E+03
TS5 (Collab No Vis)	238.9209	1.02E+04	191.9091
TS6 (Prop)	1.76E+03	8.56E+05	1.41E+03
TS6 (Collab)	5.9715	31.7045	57.1471

Table 10: Table comparing the average RSS, NEES, and GPZ values for each test scenario. Green indicates better performance for that test.

This chapter has also demonstrated the quality of the real ranging radio data. The real ranging radio data has some more noise the the simulated data, yet the overall performance of the filter is almost identical when the real data is used versus the simulated data. Additionally, the use of ranging radios greatly improves FDE capabilities

and constrains the position errors in various jamming and/or spoofing events. Yet, SOM has troubles when experiencing simultaneous sensor jamming and/or spoofing events and when there are less than two L1 satellites present.

One of the last big takeaways from this research is the unique ways in which real, asynchronous data affects ARMAS-SOM in comparison with previous simulated tests. Due to the sporadic nature of the reported radio ranges and occasional L1 satellite outages, there were some temporary negative effects on the performance of ARMAS-SOM. These negative effects include temporary instability of SOM and temporary decrease in overall performance of the ARMAS filter. Additional and lasting effects were also seen in SOM's performance for regions with two or less L1 satellites and when multiple satellites were jammed simultaneously. In these scenarios, SOM falsely indicated stability when it is clear that stability is non-existent. These oddities within SOM must be extensively analyzed in future work, and is one of the biggest unknowns at this point.

V. Conclusion

Through testing the Autonomous and Resilient Management of All-Source Sensors (ARMAS)-Stable Observability Monitoring (SOM) framework with real data from various sensors and in multiple test scenarios, the main research goals of this thesis have been accomplished. The ARMAS-SOM framework was demonstrated to be an effective all-source sensor framework for real asynchronous data, not just synchronous simulated data. It was shown that the introduction of real data produces ramifications not present in simulated scenarios. For instance, temporary sensor dropouts and occasional noise spikes occur in real data.

These features accompanying the real data, namely the temporary sensor dropouts and noise spikes, temporarily affect the performance of the filter and increase the overall filter covariance. These elements also increase the likelihood of a false alarm. Nonetheless, the ARMAS-SOM framework is still able to detect single-simultaneous sensor failures and add additional sensors when observability is threatened (although this function needs improvement).

Furthermore, the capabilities of ARMAS were stretched beyond their limits. For example, the removal of each satellite one by one in Test Scenario 3 eventually results in a loss of observability. Similarly, the addition of a bias to each satellite in Test Scenario 5 tests compares the Fault Detection and Exclusion (FDE) capabilities of ARMAS between the collaborative (with ranging radios) and proprioceptive (without ranging radios) instances of ARMAS. This scenario primarily shows the effectiveness of collaboration in FDE, but also verifies the ineffectiveness of FDE when observability is lost.

5.1 Future Work

5.1.1 Improving the Ranging Radio Data

Although the main research goals of this thesis have been accomplished, there is much future research to be done in this area. For one, it would be helpful to improve the ranging radio data. One way this could be done is by adding more ranging radios into the system. It is clear from the collaborative tests that more available ranges improves the results. This is expected, as an increase in available information should theoretically improve the results. It would also be helpful to experiment in changing the orientation of the antennas, in particular on the aircraft, and see if this helps make the ranging conversations occur more consistently. As these test scenarios show, ranging radio measurement dropouts result in temporary degradation in the overall filter performance. Modifying the orientation of the antennas on board the aircraft is one potential way to minimize these temporary measurement dropouts.

In addition to increasing the number of radios and increasing the efficiency of the setup, it would also be helpful to test different parameters within ARMAS to see if these changes would improve the solution. More experiments with the covariance values and filter thresholds could help determine the ideal values for optimal performance.

5.1.2 Other Areas for Improvement

The aforementioned potential areas for future work focused on improving the ranging radio data. However, since the goal of this research is to test ARMAS-SOM with more real data, adding more sensors would also be very helpful. In these simulations, the only sensors consisting of exclusively real data was the L1 and ranging radio sensors. The vision-based sensor consisted of a mix of real and simulated data while the velocity sensor contained exclusively simulated data. Adding various ve-

locity, position, and other real sensors would go a long way towards improving the consistency of the navigation solution and ensure there are enough sensors in order to maintain system integrity. Adding more real sensors would also further verify that ARMAS-SOM is effective when employed in real applications.

One last important point to make concerning this data is that these test runs were not completed during the actual flight. This thesis utilized post-processed real data. Real time utilization is an area where much improvement can be made. A couple of hurdles have been jumped in this thesis to compensate for real data. For example, allowing for asynchronous measurements and adding more real sensors is a big step forward; however, there are multiple other areas which need improved to allow for real-time applications of the ARMAS-SOM framework.

SOM and the logic which determines when to add collaborative information is likely the most urgent area in need of improvement. SOM was demonstrated to work quite well with simulated data and worked acceptably in some instances with the real data, yet the room for improvement is vast. For example, there are multiple instances where SOM indicates stable observability when there is clearly not enough sensor information. This results in situations where collaboration is needed, but no sensors are added due to the observability threat remaining undetected. Moreover, there are other instances where temporary unstable SOM periods cause collaborative information to be added prematurely. This is not as much of a concern as SOM falsely indicating stability for extended periods of time, nonetheless the logic can most certainly be improved to avoid adding collaborative information too soon.

Another necessary addition is the ability to have the ranging radios include the position data with the range response. Right now, the only data being received by the requester radio is the ranging data. The position of the radio at the time of range was added after the fact in the code. This is a doable addition to the code, as

the P440 radios are capable of sending a data packet along with the range response. Therefore, as long as the P440 is connected to a computer and knows its position, it will be able to update its current position to the data buffer and have this sent with the ranging packet. This addition will decrease the update rate of the ranges, but this will likely have little effect on the navigation solution due to it only being a nominal change in the update rate. Additionally, in order to implement this in real-time, it would be helpful to have ARMAS running in Python. This current version utilizes MATLAB, however Python is more versatile in operating on different computers and in real-time.

In conclusion, this research has made some important steps in demonstrating that ARMAS is effective not only with simulated data but also with real data. These test scenarios were produced using real L1 data, real ranging radio data, real/simulated mix of vision-based position data, and simulated velocity data. While there is still room for improvements to the current state of ARMAS-SOM (i.e. using only real data, removing SOM abnormalities, etc.), this research paves the way for testing ARMAS with even more real data. In this research, ARMAS has demonstrated its effectiveness in handling multiple sensors producing real and asynchronous data. For this reason, it is reasonable to conclude that the ARMAS-SOM framework is capable of being used in real-time with real sensors.

Bibliography

1. Juan Jurado and John Raquet. Autonomous and resilient management of all-source sensors. *ION*, pages 142–159, 2019.

2. Jonathon Gipson and Robert Leishman. Resilient collaborative all-source navigation. *IEEE*, 2021.

3. Andrew Appleget, Robert Leishman, and Jonathon Gipson. Evaluation of sensor-agnostic all-source residual monitoring for navigation. *ION*, pages 339–353, 2021.

4. Jonathon Gipson and Robert Leishman. Resilience for multi-filter all-source navigation framework with integrity. *Afit Scholar*, 2021.

5. Aaron Canciani and John Raquet. Absolute positioning using the earth's magnetic anomaly field. *Navigation*, 63(2):111–126, 2016.

6. Mitchell C. Hezel. Improving aeromagnetic calibration using artificial neural networks. Master's thesis, AFIT, WPAFB, OH, Mar 2020.

7. Kyle A. Emery. Modeling aircraft disturbance fields for magnetic navigation using dense anns and the novel manntl architecture. Master's thesis, AFIT, WPAFB, OH, Mar 2021.

8. D.T. Venable. *Improving Real World Performance of Vision Aided Navigation in a flight environment.* PhD thesis, AFIT, WPAFB, OH, September 2016.

9. Joseph Curro and John Raquet. Navigation using vlf environmental features. In *Position, Location, and Navigation Symposium (PLANS)*, 2016 IEEE/ION, pages 373–379. IEEE, 2016.

10. Jonathon Gipson. *An Enhanced Framework for Collaborative All-Source Navigation with Integrity*. PhD thesis, AFIT, WPAFB, OH, August 2021.

11. TDSR. *RangeNet User Guide*. TDSR, Petersburg, TN, 2020.

12. TDSR. *Datasheet / User Guide P440 UWB Module*. TDSR, Petersburg, TN, 2020.

13. ENGIBEX. Antennas for dummies. `https://engibex.com/antennas-for-dummies/`. Accessed: 2021-12-20.

14. Time Domain. *BroadSpec UWB Antenna Data Sheet*. Time Domain, Huntsville, AL, 2017.

15. Time Domain. *RangeNet Programming Application Interface (API) Specification*. Time Domain, Petersburg, TN, 2020.

16. Time Domain. *Guidelines for Augmenting the Broadspec Antenna with a Planar Back-Reflector*. Time Domain, Huntsville, AL, 2016.

17. Juan Jurado. *Autonomous and Resilient Management of All-Source Sensors for Navigation Assurance*. PhD thesis, AFIT, WPAFB, OH, September 2019.

18. W.R. Michalson. Ensuring gps navigation integrity using receiver autonomous integrity monitoring. *IEEE Aerospace and Electronic Systems Magazine*, 10(10):31–34, 1995.

19. B.W. Parkinson and P. Axelrad. Autonomous gps integrity monitoring using the pseudorange residual. *Navigation*, 35(2):255–274, 1988.

20. R.G. Brown and P. McBurney. Self-contained gps integrity check using maximum solution separation. *Navigation*, 35(1):41–53, 1988.

21. T. Walter and P. Enge. Weighted raim for precision approach. In *Proceedings of ION GPS*, volume 8, pages 1995–2004. Institute of Navigation, 1995.

22. M. Cai E. Wang and T. Pang. A simple and effective gps receiver autonomous integrity monitoring and fault isolation approach. In *Control Engineering and Communication Technology (ICCECT)*, 2012 International Conference, pages 657–660, 2012, 2012. IEEE.

23. S.-Y. R. Young and G.A. McGraw. Method and system for fault detection and exclusion for multi-sensor navigation systems. US Patent 7,219,013, May 15 2007.

24. P.Y. Hwang and G.A. McGraw. Receiver autonomous signal authentication (rasa) based on clock stability analysis. In *Position, Location, and Navigation Symposium -PLANS 2014*, 2014 IEEE/ION, pages 270–281. IEEE, 2014.

25. M. Joerger and B. Pervan. Kalman filter-based integrity monitoring against sensor faults. *Journal of Guidance, Control, and Dynamics*, 36(2):349–361, 2013.

26. M. Joerger L. Gratton and B. Pervan. Carrier phase relative raim algorithms and protection level derivation. *The Journal of Navigation*, 63(2):215–231, 2010.

27. J. Neale M. Joerger and B. Pervan. Iridium/gps carrier phase positioning and fault detection over wide areas. In *Proceedings of the 22nd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2009)*, pages 1371–1385, 2009.

28. J. Podobnik T. Beravs, S. Begus and M. Munih. Magnetometer calibration using kalman filter covariance matrix for online estimation of magnetic field orientation. *IEEE Transactions on Instrumentation and Measurement*, 63(8):2013–2020, 2014.

29. Z. Wang Z. Wu and Y. Ge. Gravity based online calibration for monolithic triaxial accelerometers' gain and offset drift. In *Intelligent Control and Automation*,

volume 3 of *Proceedings of the 4th World Congress*, pages 2171–2175. IEEE, 2002.

30. J. Collin M.Kirkko-Jaakkolam and J.Takala. Bias prediction for mems gyroscopes. *IEEE Sensors Journal*, 12:2157–2163, 2012.

31. J. Levinson and S. Thrun. Automatic online calibration of cameras and lasers. In *Robotics: Science and Systems*, volume 2, 2013.

32. K.Nakamura H. Miura, T. Yoshida and K. Nakadai. Slam-based online calibration of asynchronous microphone array for robot audition. In *Intelligent Robots and Systems (IROS)*, 2011 IEEE/RSJ International Conference, pages 524–529. IEEE, 2011.

33. J.D. Jurado and C.C. McGehee. Complete online algorithm for air data system calibration. *Journal of Aircraft*, pages 1–12, 2018.

34. N.Keivan and G. Sibley. Constant-time monocular self-calibration. In *International Conference on Robotics and Biomimetics*, pages 1590–1595, 2014.

35. Z. Yang and S. Shen. Monocular visual inertial fusion with online initialization and camera-imu calibration. In *SSR 2015 - 2015 IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2016.

36. C. Jia and B.L. Evans. Online calibration and synchronization of cellphone camera and gyroscope. In *2013 IEEE Global Conference on Signal and Information Processing*, GlobalSIP 2013 - Proceedings, pages 731–734, 2013.

37. Y. Ling and S. Shen. High-precision online markerless stereo extrinsic calibration. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1771–1778. IEEE, 2016.

38. S. Diverdi and J.T. Barron. Geometric calibration for mobile, stereo, autofocus cameras. In *2016 IEEE Winter Conference on Applications of Computer Vision*, WACV 2016, 2016.

39. N. Zhou S. Akhlaghi and Z.Huang. Adaptive adjustment of noise covariance in kalman filter for dynamic state estimation. In *Power and Energy Society General Meeting*, 2017 IEEE, pages 1–5, 2017.

40. B.J. Etherton C.H. Bishop and S.J. Majumdar. Adaptive sampling with the ensemble transform kalman filter. part i: Theoretical aspects. *Monthly Weather Review*, 129(3):420–436, 2001.

41. A.S. Lawless J.A. Waller, S.L. Dance and N.K. Nichols. Estimating correlated observation error statistics using an ensemble transform kalman filter. *Tellus A: Dynamic Meteorology and Oceanography*, 66(1):23294, 2014.

42. C.B. Chang and M. Athans. Hypothesis testing and state estimation for discrete systems with finite-valued switching parameters. Technical report, Massachusetts Inst of Tech Cambridge Electronic Systems Lab, 1977.

43. P. Eide and P. Maybeck. An mmae failure detection system for the f-16. *IEEE Transactions on Aerospace and Electronic Systems*, 32(3):1125–1136, 1996.

44. P.D. Hanlon and P.S. Maybeck. Multiple-model adaptive estimatino using a residual correlation kalman filter bank. *IEEE Transactions Aerospace Electronic Systems*, 36(2):393–406, 2000.

45. Juan Jurado et al. Residual-based multi-filter methodology for all-source fault detection, exclusion, and performance monitoring. *ION*, pages 493–510, 2019.

46. Christoffer Heckman Simon Julier Chen, ZhaoZhong and Nisar Ahmed. Weak in the nees?" auto-tuning kalman filters with bayesian optimization. *FUSION 2018*, pages 1072–1079, 2018.

Acronyms

AFIT Air Force Institute of Technology. 30

ANT Autonomy and Navigation Technology. 30

ARMAS Autonomous and Resilient Management of All-Source Sensors. iv, vi, vii, viii, xi, 1, 2, 3, 4, 5, 7, 8, 9, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 29, 30, 37, 38, 44, 45, 47, 49, 50, 51, 52, 53, 56, 61, 63, 64, 68, 76, 78, 80, 86, 89, 91, 92, 93, 94, 95

CAT Channel Analysis Tool. 10

ENU East-North-Up. viii, ix, 23, 31, 38, 39, 44, 46, 48, 49, 56, 57, 61

FDE Fault Detection and Exclusion. iv, 1, 4, 5, 18, 21, 24, 25, 26, 58, 68, 71, 80, 82, 86, 90, 92

GNSS Global Navigation Satellite System. iv, 1, 2, 7, 8, 16, 29, 31, 36, 37

GPS Global Positioning System. iv, vii, xi, 2, 21, 24, 32, 36, 37, 38, 40, 48, 49, 50, 61

GPZ Guaranteed Position Zone. viii, ix, x, 22, 24, 28, 58, 59, 68, 69, 70, 71, 72, 73, 74, 75, 79, 84, 85, 90

IMU inertial measurement unit. 7, 16, 32

LLA latitude-longitude-altitude. 36, 38

MRM Monostatic Radar Module. 10

NEES Normalized Estimation Error Squared. ix, x, 22, 23, 24, 29, 53, 55, 58, 66, 67, 68, 69, 70, 71, 75, 78, 79, 82, 83, 87, 88, 89, 90

PII Pulse Integration Index. 10, 35

PR Pseudorange. 38, 39

PRM Precision Range Measurement. 10, 49

PTP Precision Time Protocol. 32, 35, 36

RN RangeNet. 10

RSS Root Sum of Squares. viii, ix, x, 22, 23, 29, 53, 54, 62, 63, 64, 65, 66, 68, 69, 71, 75, 76, 77, 79, 80, 81, 87, 88, 90

SAARM Sensor-Agnostic All-Source Residual Monitoring. vi, 16, 18, 21, 24, 25

SOM Stable Observability Monitoring. iv, vi, 1, 2, 3, 4, 5, 7, 8, 9, 16, 18, 21, 22, 23, 24, 25, 26, 29, 30, 37, 47, 51, 52, 53, 55, 56, 59, 60, 61, 62, 75, 76, 78, 79, 80, 85, 86, 87, 89, 91, 92, 93, 94, 95

TDMA Time Division Multiple Access. vi, 10, 33

TS Test Scenario. viii, ix, x, 52, 53, 54, 55, 78, 85, 86, 89, 90

UAV unmanned aerial vehicle. 2, 5, 25, 27, 32

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 — Mar 2022

4. TITLE AND SUBTITLE

EVALUATION OF THE ARMAS-SOM FRAMEWORK WITH REAL DATA

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Brandon M. Blakely

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-009

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

AFWERX
WPAFB OH 45433-7765
Email: afrl.pa.inquiry@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFWERX

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

The ARMAS framework was created with the goal of developing a framework for all-source sensors which is able to combine detection, identification, calibration, model selection, and independent evaluation into a single system. Stable Observability Monitoring (SOM), augments the original ARMAS framework by enabling ARMAS to detect whether or not its Fault Detection and Exclusion (FDE) capabilities can be trusted and when additional sensor information is required to maintain resiliency. While previously tested with simulated sensor data, SOM has yet to be tested with real-life sensor data. Furthermore, ARMAS has only been tested with real-life GNSS data. This work expands on previous work by testing the ARMAS-SOM framework with real-life GPS, ranging radio, and camera sensor data, while analyzing its strengths and weaknesses.

15. SUBJECT TERMS

all-source, autonomous, Autonomous and Resilient Management of All-Source Sensors (ARMAS),Fault Detection and Exclusion (FDE), GNSS-denied, navigation, ranging radios, UAV, Vision-Based

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Robert C. Leishman, AFIT/ENG
U	U	U	UU	115	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636, ext 4755