



Graduate Theses, Dissertations, and Problem Reports

2018

IMPLEMENTATION OF A STATE-OF-THE-ART GNSS RECEIVER AUTONOMOUS INTEGRITY MONITORING TECHNIQUE

Shannen Daly

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Recommended Citation

Daly, Shannen, "IMPLEMENTATION OF A STATE-OF-THE-ART GNSS RECEIVER AUTONOMOUS INTEGRITY MONITORING TECHNIQUE" (2018). *Graduate Theses, Dissertations, and Problem Reports*. 7171.
<https://researchrepository.wvu.edu/etd/7171>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

**IMPLEMENTATION OF A STATE-OF-THE-ART GNSS RECEIVER
AUTONOMOUS INTEGRITY MONITORING TECHNIQUE**

Shannen Daly

Thesis submitted to the
Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Aerospace Engineering

Jason Gross, Ph.D., Chair

Evan Anzalone, Ph.D.

Patrick Browning, Ph.D.

Yu Gu, Ph.D.

Department of Mechanical and Aerospace Engineering

Morgantown, West Virginia

2018

Keywords: GPS, Galileo, GNSS, RAIM, Aviation

Copyright 2018

Abstract

Implementation of a State-of-the-Art GNSS Receiver Autonomous Integrity Monitoring Technique

Shannen Daly

This thesis implements a state-of-the-art solution separation advanced RAIM (ARAIM) algorithm as it is written as reported in the literature. Specifically, a GNSS fault detection and exclusion algorithm for a multi-constellation GNSS was implemented in software and tested against simulated data. RAIM algorithms have been created in many forms over the last couple of decades and are still in development today. The position solution results produced by this ARAIM algorithm were compared to that of a snapshot weighted least squares (WLS) solution in which failed satellites are removed before processing and an WLS solution with no corrections applied. In addition, the difference in position solution between ARAIM and the simulation truth was compared to the ARAIM reported horizontal and vertical protection limits, as well as, the position performance criteria. This thesis also investigates the performance of the exclusion method and how it affects the performance of the overall ARAIM algorithm. The algorithm implemented and tested in this thesis will be used as a basis of comparison for on-going research into robust GNSS processing techniques.

Acknowledgements

I would like to thank the NASA West Virginia Space Grant Consortium (WVSGC) for providing funding for this thesis research through the NASA WV Space Grant Consortium Director's Discretionary Fund.

Table of Contents

Abstract	ii
Acknowledgements	iii
List of Tables	vi
List of Figures	vii
Important Terms.....	viii
Chapter 1 Introduction	1
1.1 Motivation	1
1.2 Thesis Outline.....	2
Chapter 2 Literature Review.....	3
2.1 Receiver Autonomous Integrity Monitoring (RAIM)	3
2.1.1 Modern RAIM Techniques	4
2.1.1.1 Solution Separation RAIM Techniques.....	5
2.1.2 Multiple Constellation RAIM.....	7
Chapter 3 Background	8
3.1 Performance Characteristics	8
3.1.1 Integrity	8
3.2 Global Positioning System and Galileo Global Navigation Satellite Systems	9
3.3 Weighted Least Squares	10
Chapter 4 Technical Approach.....	11
4.1 Algorithm Overview.....	11
4.2 Fault Detection.....	18
4.2.1 Minimum Chi-Squared Statistic.....	18
4.2.2 Rank-one Updates	19
4.3 Fault Exclusion.....	19
Chapter 5 Testing.....	20
5.1 Sensitivity Analysis	21
5.1.1 Chi-Square Statistic	22
5.1.2 Increasing Potential Fault Modes	22
Chapter 6 Results	23
6.1 Position Error Comparison	23
6.2 Fault Detection and Exclusion Capability	28

6.3 Position Performance Measurements.....	33
6.4 Increasing Potential Fault Modes.....	34
Chapter 7 Conclusions and Future Work.....	35
7.1 Conclusions.....	35
7.2 Future Work.....	36
References.....	37

List of Tables

Table 0.1: Technical variables and their significance used in the ARAIM algorithm as defined in [1] and [2].	viii
Table 6.1: Minimum and maximum HPL and VPL and maximum position difference between ARAIM and truth.	25
Table 6.2: Root mean square and mean of the difference in positioning data between ARAIM and the truth.	25
Table 6.3: Results of fault exclusion by varying injected fault size.	32
Table 6.4: ICAO SARPs	33
Table 6.5: Performance of ARAIM in each test run for ten-meter injected errors	33

List of Figures

Figure 4.1: Block diagram showing how the ARAIM algorithm runs through each step.....	12
Figure 5.1: Trajectory of simulated flight.	21
Figure 6.1: Position error for ARAIM in the east direction with HPL and markers for injected error.	24
Figure 6.2: Position error for ARAIM in the north direction with HPL and markers for injected error.	25
Figure 6.3: Position error for ARAIM in the vertical direction with VPL and markers for injected error.	25
Figure 6.4: Overlay of the difference in positioning of ARAIM and WLS and difference in positioning of simulation truth and WLS in the X or east direction.....	27
Figure 6.5: Overlay of the difference in positioning of ARAIM and WLS and difference in positioning of simulation truth and WLS in the y or north direction.	27
Figure 6.6: Overlay of the difference in positioning of ARAIM and WLS and difference in positioning of simulation truth and WLS in the z or vertical direction.	28
Figure 6.7: Chi-Squared Statistic Values calculated for each fault mode in one epoch before exclusion of the failed satellite.	29
Figure 6.8: Graphs depicting an overlaying of the difference in position solution between ARAIM and WLS with corrections, ARAIM and WLS with no corrections, and WLS with corrections and without. The graphs are magnified to focus on the difference between the three position solutions and represent two spans of time within the same test period. The purple points on the graph show where there is an injected failure.	30
Figure 6.9: Graph depicting an overlaying of the difference in position solution between ARAIM and WLS with corrections, ARAIM and WLS with no corrections, and WLS with corrections and without in the east. The graph is magnified to focus on the difference between the three position solutions and represent a span of time within the larger test period. The purple points on the graph show where there are errors injected.	31
Figure 6.10: Graph depicting an overlaying of the difference in position solution between ARAIM and WLS with corrections, ARAIM and WLS with no corrections, and WLS with corrections and without in the north. The graph is magnified to focus on the difference between the three position solutions and represent a span of time within the larger test period. The purple points on the graph show where there are errors injected.	31
Figure 6.11: Graph depicting an overlaying of the difference in position solution between ARAIM and WLS with corrections, ARAIM and WLS with no corrections, and WLS with corrections and without in the vertical. The graph is magnified to focus on the difference between the three position solutions and represent a span of time within the larger test period. The purple points on the graph show where there are errors injected.	32
Figure 6.12: The amount of fault modes needed for consideration of increasing number of possible simultaneous faults from 2 to 3. The graph on the left shows the increase from 1 to 2 simultaneous faults and the graph on the right shows the increase from 1 to 3 simultaneous faults.	35

Important Terms

Table 0.1: Technical variables and their significance used in the ARAIM algorithm as defined in [1] and [2].

$b_{nom,i}$	Maximum nominal bias for satellite i as given by the Integrity Service Message.
C_{acc}	Nominal error model for accuracy.
C_{int}	Nominal error model for integrity.
e_q	A vector with the q th value being one and all others zero.
G	Geometry matrix.
HPL	Horizontal Protection Limit.
K_{ACC}	Number of standard deviations used in accuracy formula, coefficient multiplied to standard deviation of vertical position.
$K_{fa,q}$	Coefficient of probability of fault and number of fault modes based upon direction. Used to calculate threshold used in integrity testing.
K_{FF}	Number of standard deviations used in fault-free vertical position error calculation, coefficient multiplied to standard deviation of vertical position.
$P_{Const,not\ monitored}$	Probability that a constellation of satellites is not being monitored in that epoch.
P_{FA_HOR}	The amount of the continuity budget allowed for the horizontal directions.
P_{FA_VERT}	The amount of the continuity budget allowed for the vertical direction.
$P_{Sat,not\ monitored}$	Probability that a satellite is not being monitored in that epoch.
$PHMI_{HOR}$	The amount of the integrity budget allowed for the horizontal directions.
$PHMI_{VERT}$	The amount of the integrity budget allowed for the vertical direction.
$T_{k,q}$	Threshold value calculated for use in integrity test during each epoch, for each fault mode k , and direction q .
W	Weight matrix.
y	Pseudorange measurement.
$\sigma_{URA,i}$	Standard deviation of the satellite clock and ephemeris error for integrity, used in building weight matrix for least squares solution in horizontal directions.
$\sigma_{URE,i}$	Standard deviation of the satellite clock and ephemeris error for accuracy and continuity, used in building weight matrix for least squares solution in vertical direction.
$\sigma_{v,acc}$	Standard deviation for the vertical position solution.
accuracy(95%)	The value under which the position error will be lower than ninety-five percent of the time.
fault free (10^{-7})	Value that if a result is beneath this, guarantees fault free accuracy of the position solution.
EMT	Effective Monitor Threshold; Maximum detection threshold that increases if the probability of fault increases above a set level for a certain epoch.
VPL	Vertical Protection Limit.

Chapter 1 Introduction

1.1 Motivation

The Global Positioning System (GPS) was developed in the 1960s when multiple United States organizations became interested in three-dimensional world-wide position determination [3]. This interest grew to include use in aviation and became officially used in civil aviation when the first GPS unit was approved for civil flight operation in 1994 [4]. The approval of this GPS receiver came with regulations and standards that must be maintained, of which included integrity or to what level the measurements made can be trusted to be correct [5]. Receiver Autonomous Integrity Monitoring (RAIM), the method used to determine the integrity of a position solution, was used in operation of aircraft in the mid-1990s [1]. Initially this technique was used in steady flight, however as time went on and technology advanced to higher capabilities, RAIM has been considered for use in more demanding flight instances. These flight instances, mainly consisting of precision approaches and landing, have tighter requirements for navigation error and subsequent horizontal and vertical protection limits [1]. Galileo and GLONASS, two satellite constellations created by European and Russian governments, are capable of being used with the GPS constellation in a larger Global Navigation Satellite System (GNSS) [6]. An Advanced RAIM (ARAIM) method that builds an Integrity Support Message (ISM) and broadcasts it to aircraft was developed by the Working Group C ARAIM Technical subgroup as written in [7]. Previous to the writing of [1], the ARAIM method has been shown to only cover single-fault cases and multiple simultaneous faults would have to be resolved by the airborne receiver [1]. The additions to the algorithm written in [1] add the capabilities of multi-constellation use, correction of multiple fault cases, and more accuracy in the vertical direction. The use of multiple constellations also adds the possibility of constellation-wide

faults in which an entire constellation of satellites would have to be excluded and thus must be considered as well.

This thesis implements an ARAIM user algorithm as it is written in [1]. During this study, a software class written in Python with the baseline ARAIM algorithm was further developed to include methods that designate when a failure is present, identify the failed satellite, and exclude all data associated with said satellite for the epoch in which the failure was detected. The algorithm implemented in this thesis is directly from [1] as it is the current ARAIM user algorithm. The implementation of this algorithm is tested against simulated data with an assumed ISM, but will be tested further with more simulated data and flight data when it is used for on-going research into robust GNSS processing techniques as a basis for comparison [8] [9].

1.2 Thesis Outline

Chapter 2 of this thesis contains a review of the technical literature about the development of RAIM including information about modern RAIM and multi-constellation RAIM. The third chapter provides a background on additional topics that pertain to RAIM such as an overview of GPS, flight characteristics with an emphasis on integrity, and a derivation of least squares with an extension to weighted least squares. Chapter 4 discusses the technical approach taken in this study and the details of the ARAIM algorithm implemented in this study as adopted from [1]. The fifth chapter contains a discussion on how the ARAIM algorithm was tested with simulation data. Results are discussed in chapter six, including a comparison of the positioning results to the truth and a simple weighted least squares approach as well as the performance of the solution separation ARAIM algorithm in detecting failures. The final chapter presents the conclusions drawn from this study as well as future work.

Chapter 2 Literature Review

2.1 Receiver Autonomous Integrity Monitoring (RAIM)

This section provides a brief review of weighted least squares, insight into the overall process of RAIM, a discussion on modern state-of-the-art RAIM methods, and the capability of RAIM over multiple constellations of satellites.

Linear least squares is a technique in which a maximum likelihood estimation is made for a vector of values that may be fit to a model using a linear model [10]. This technique is often used to calculate a position solution using the coordinate location and pseudorange data for in-view satellites. A weight may be applied that is dependent upon factors unique to each satellite. This weight, factored into the solution through use of a weight matrix, brings into account various sources of potential error or noise. With the formation of the all-in-view solution, the solution in which all available satellites in view are used to form a position solution, a few different additional quantities may be created that can be used to check the health and integrity of the satellite data.

RAIM is a technique utilized to ensure the quality of satellite data being used to form the position solution of aircraft. This technique was first known as a different name, multiple hypothesis solution separation. The term RAIM wasn't formally used until 1987, it was seen in literature as early as 1986 at an ION meeting [11]. Moving forward from these initial studies, the basis of modern RAIM techniques can be observed in [12]. Of these ideas that have been carried into state-of-the-art techniques are the comparison of various combinations of satellites, similar to solution separation, and the use of the residual and covariance.

In the early development years of RAIM, comparison of redundant measurements was the state-of-the-art approach to integrity monitoring as seen in [13] and [14]. The redundant measurements refer to having more satellites than minimum necessary. A full position solution

requires four satellites to be seen as a minimum and there are typically more available, allowing for multiple sets of in-view satellites to be compared. This comparison approach could detect errors in the pseudorange at the one-hundred-meter level [13]. Development of various schemes persisted over a few years until the release of the GPS Minimum Operational Performance Standards (MOPS) by the Radio Technical Commission for Aeronautics (RTCA) in the early 1990s, which brought with it integrity requirements for GPS receiver performance. The release of these standards brought forth a need for a standard base algorithm as proposed in [15]. This basic technique, the equation for which is shown below, centered the solutions formed by three RAIM techniques around the measurements described by an overdetermined system of linear equations in the form below where \mathbf{G} is the geometry matrix of the satellite positions, \mathbf{x} are the three components of the measured user position from nominal, \mathbf{y} is the difference in measured and computed pseudorange, and $\boldsymbol{\epsilon}$ is measurement error due to factors such as noise. The three techniques that were unified by this new base measurement system were the range comparison, least squares residual, and parity space methods [15].

$$\mathbf{y} = \mathbf{G}\mathbf{x} + \boldsymbol{\epsilon} \quad (1)$$

2.1.1 Modern RAIM Techniques

As years of research went on into RAIM and the related methods, the MOPS developed and changed as well. One of the major changes that affected what RAIM algorithms were used was the inclusion of a protection radius. The protection radius is a particular bound calculated in which the position solution may be guaranteed to be within. Since this performance index uses the sum-of-the-squared errors (SSE), the least squares residual and parity space methods became the mainly used method at the time [16]. The SSE is a term that may be formed as a result of the least squares approach used to form the position solution. It is calculated through use of the error present in the

position solution formed, represented by the variable w , shown below in equations 2 and 3 [11], and provides a few advantages when considering failure detection. Those advantages include providing an easy determinate for failure as it is a nonnegative scalar quantity [11]. This change in preference is also due in part to all three of the aforementioned methods being equivalent as per their shared base measurement system as previously described in [15].

$$w = [I - G(G^T G)^{-1} G^T]y \quad (2)$$

$$SSE = w^T w \quad (3)$$

2.1.1.1 Solution Separation RAIM Techniques

For a majority of the 1990s and the early to mid-2000s least squares residual and parity space methods were considered the state-of-the-art RAIM methods. Kalman filtering and integrating GPS with other sensors, such as visual cameras [17] and inertial measurement units in strapdown inertial navigation systems were considered for developing and maintaining high integrity navigation systems [18] [19]. These techniques integrated the use of radar imagery [17] and presented a new approach to detecting failures through use of comparing the chi-square statistic of the state estimate of the Kalman filter and the formed values that are consequent of finding the position solution [19]. Desired performance grew to include multiple simultaneous faults, the emergence of new GNSS satellite constellations, and protection in both the horizontal and vertical directions. The development and further use of the Wide Area Augmentation System (WAAS) in which information from ground GPS receivers are used to augment GPS data and deliver corrections as well as necessary integrity messages and alerts [20], was intended to help cover these new requirements in performance. WAAS implementation is desired to be used in precision navigation scenarios, such as

precision approaches and landing [20], and thus have stricter requirements for navigation in the horizontal and vertical [1]. This new requirement of protection in the vertical spurred the development of multiple new RAIM techniques around the mid-2000s and included *NIORAIM* in [21], Optimally Weighted Average Solution (OWAS) in [22], and Multiple Hypothesis Solution Separation (MHSS) initially in [23] and more recently in [24]. Equipment designed to further support the improvement of RAIM performance, particularly in achieving successful high precision navigation scenarios, was developed in the form of integrity beacons. These beacons could act as pseudolites and when placed in the path leading to the landing zone for an aircraft, can support the auto-landing of said aircraft [25]. An algorithm to detect multiple simultaneous faults, the Range Consensus (RANCO) algorithm, was developed in a cooperative effort between Stanford University and the German Aerospace Center. This algorithm calculates a position solution based upon the data of four available satellites and compares it to the pseudoranges of all available satellites that were not used in the initial position solution [26]. Using an approach similar to that of the Random Sample Consensus Algorithm (RANSAC), all possible subsets of four satellites are taken into account and the difference in position solution between the four satellite solution and the extra satellites are calculated and compared to result in the solution set that does not exceed the testing threshold [26]. Although promising, this algorithm still needs to be further investigated for error correlation and generalization of techniques [27].

The solution separation method was known in a different but not widely practiced form in previous years according to [11] due to the increased difficulty in mathematically analyzing its effectiveness in comparison to the residual based methods. It has been optimized for improvements in simplicity of protection level calculation and the use of multiple range sources to become known as multiple hypothesis solution separation RAIM in [24]. In the solution separation method of RAIM, ‘fault modes’ are formed by removing one or two satellites with rank-one updates and the

position solution that is created as a result is compared to that of the all-in-view solution. When a failure is detected through a fault mode not passing the threshold test, the satellite that corresponds to the fault mode furthest from the all-in-view solution would be considered the source of the error. This will be covered in further detail within the technical discussion of chapter 4. This solution separation algorithm was settled upon as the state-of-the-art method over residual-based because of a lowered risk of an error being present due to the availability of tailored test statistics as stated in [28]. An advanced RAIM, or ARAIM, algorithm was discussed in [1] and is the basis of modern ARAIM studies as well as the basis for this thesis. One such study is shown in [29] in which the performance of said algorithm is discussed and the large computational load of ARAIM is discussed and possible improvements suggested.

2.1.2 Multiple Constellation RAIM

Another point of discussion regarding the performance and capability of RAIM is that of the use of multiple constellations when forming position solutions. This idea was first seen in 1992 and was proposed by [30] with GPS and GLONASS being the two constellations in mind for use. The choice in constellations changed to include two options in addition to GPS with the emergence of the new Galileo constellation in the early 2010s. The addition of this new GNSS constellation raised the need of investigating integrity monitoring techniques, such as RAIM, use of ground based integrity monitoring stations, and a new method of satellite autonomous integrity monitoring (SAIM), in a multi-constellation system [31]. This idea was extended to improve worldwide coverage that will improve the availability of RAIM to more airspace and further improve vertical coverage by RAIM in [7]. Multi-constellation satellite use is seen in [1] and is continued to be used in [29] and [32].

Chapter 3 Background

3.1 Performance Characteristics

A higher capability for performance is required in flight and navigation in instances of low coverage or when there is a necessity for accurate real-time information and response [1]. There are four required navigation performance parameters that were developed by the International Civil Aviation Organization (ICAO). They are in some way connected to each other and are: accuracy, continuity, availability, and integrity [5]. Accuracy is the degree of conformance of the position solution to the true position at that point in time [5]. Availability is the measured percentage of time that the services of the system are usable by the user of the navigation system [5]. Integrity will be defined in the next section and continuity is a measure of the entire system to provide information on the position of a craft without interruption during the duration of operation [5]. These definitions are provided in further detail in Appendix B of [5]. The particular definitions and specifics about these parameters were set forth in RTCA/DO-208 'Minimum Operational Performance Standards for Airborne Supplement Navigational Equipment Using GPS'.

3.1.1 Integrity

An important parameter of performance that is connected directly to the faith one can put into position measurements is integrity. Integrity is the measure of trust that can be placed in the correctness of the information supplied by a navigation system [5]. The exact integrity requirements to be upheld depend upon the situation at hand, but are dictated by four separate integrity parameters. These parameters are: alert limit, time to alert, integrity risk, and protection level. Alert limit is defined as the error tolerance for a system, time to alert is the amount of time allowed for the error to be above the set limit, and integrity risk is the probability that there is an error present. The final parameter is the protection limit which is the bound computed to ensure that the craft was

within a certain allowed volume of space that coincides with the given integrity budget. These parameters are all taken into consideration and an integrity budget is written for particular crafts and navigation satellites. This budget is comprised of both probabilities and physical values and are applied depending upon their contribution to a particular navigation algorithm. The solution-separation RAIM algorithm focuses upon the integrity risk and protection limit parameters but keeps all four in consideration.

3.2 Global Positioning System and Galileo Global Navigation Satellite Systems

The U.S. owned radiometric tracking service utilized by many people worldwide is known as the Global Positioning System, GPS. GPS is comprised of three segments, which are space, control, and user, that have their own components and purposes. The space and control segments are two pieces of one part of the system that ultimately transmit the signals needed for use in the radiometric navigation process. The space segment is comprised of the satellites within the GPS constellation that transmit signals that are then used to make ranging measurements [33]. The control segment is supplemental to the space segment and maintains satellites in order to keep them in functioning condition [33]. A few of the control segment's uses include updating satellites' ephemeris data, clock, and almanac, monitoring their subsystem health and status, and activating spare satellites to maintain availability [33]. The user segment is comprised of the receiver which processes L-band signals from satellites in the space segment to determine position, velocity, and time [33].

The measurements made by this system come from two different sources, code and carrier phase. The measurements are used to form the pseudorange measurements; the measurements used in this and many other navigation algorithms. The geometric range or the distance from the satellite to the receiver and pseudorange may be calculated as shown below [6].

$$r = \sqrt{(x^k - x)^2 + (y^k - y)^2 - (z^k - z)^2} \quad (4)$$

$$\rho^k = r^k + c[\delta t - \delta t^k] + I_\rho^k + T_\rho^k + \epsilon_\rho^k \quad (5)$$

In the above equations, the superscript k designates the satellite is being considered and the subscript ρ signifies that the pseudorange is being calculated. In equation 4, \mathbf{x} , \mathbf{y} , and \mathbf{z} represent the position of the satellite and the user. In equation 5, the variable c represents the speed of light, δt is the clock bias, I and T represent the ionospheric and tropospheric delays affecting the signal, and ϵ represents unmodeled effects, modeling errors, and measurement errors. This calculation forms a biased and noisy measurement of r which then must be processed for a more accurate position estimation.

Galileo, created by the European Union, is a navigation satellite constellation that operates similarly to GPS [6]. The radio frequencies of both constellations, GPS and Galileo, are compatible as their signals will not interfere or degrade each other [6]. This in turn allows for them to be used together in the larger GNSS for a system that provides more coverage as more satellites will be in view and utilized for a more precise positioning solution.

3.3 Weighted Least Squares

Least squares, the basis of finding the position solution, is derived from the equation that illustrates the linear relationship between a column vector \mathbf{x} of estimated unknown parameters and \mathbf{y} , a set of noisy pseudorange measurements [10].

$$\mathbf{y} = \mathbf{G}\mathbf{x} + \boldsymbol{\epsilon} \quad (6)$$

In equation 6 above, the noisy measurements are linearly related to the estimated parameters by first multiplying them to the geometry matrix \mathbf{G} and then adding on the error added through noise $\boldsymbol{\epsilon}$. To solve for \mathbf{x} , the maximum likelihood estimate would have to be made so that a proper

estimate may be made given the observations. If it is assumed that measurement errors are Gaussian distributed and errors in other measurements statistically independent, then the maximum likelihood estimate may be solved for as it is in the following equation [10].

$$\hat{x} = \min_x \|y - Gx\|^2 \quad (7)$$

This may be solved by differentiating with respect to \hat{x} and setting the derivation to zero resulting in the following least squares estimate [10].

$$\hat{x} = (G^T G)^{-1} G^T y \quad (8)$$

If the errors present are not identically distributed or independent, as in the case of a multiple satellite constellation, then a weighting matrix will need to be included in the calculations and derivation that will result in an altered version of the least squares estimation or a weighted least-squares estimation [10].

$$\hat{x} = (G^T W G)^{-1} G^T W y \quad (9)$$

Chapter 4 Technical Approach

4.1 Algorithm Overview

The solution separation RAIM technique involves the formation of a unique fault mode for each satellite and constellation present [1]. Each fault mode is formed by taking the difference of the particular satellite's impact on the position solution with respect to the all-in-view solution. The satellite whose solution is furthest away from that of the all-in-view, if a fault is detected, is considered the failed satellite. The data of this failed satellite would then be excluded from the data for the current epoch and the all-in-view solution reformed and evaluated. Although the position solution results of the two methods, solution separation and residual based RAIM, are the same the

two methods have their fundamental differences. The solution separation method results provide a greater protection over a larger variety of failures as a test threshold, chi-squared statistic, and protection levels tailored to the data are formed [28].

RAIM algorithms are all built upon the same general idea. That idea is to detect faults within the pseudorange data received from GPS satellites. The particular algorithm used is written to follow that of the one presented in the paper *Baseline Advanced RAIM User Algorithm and Possible Improvements* by Blanch, et al [1]. The following block diagram shows how the algorithm runs and how the results from the previous step and values from the ISM and set budgets flow into the next step.

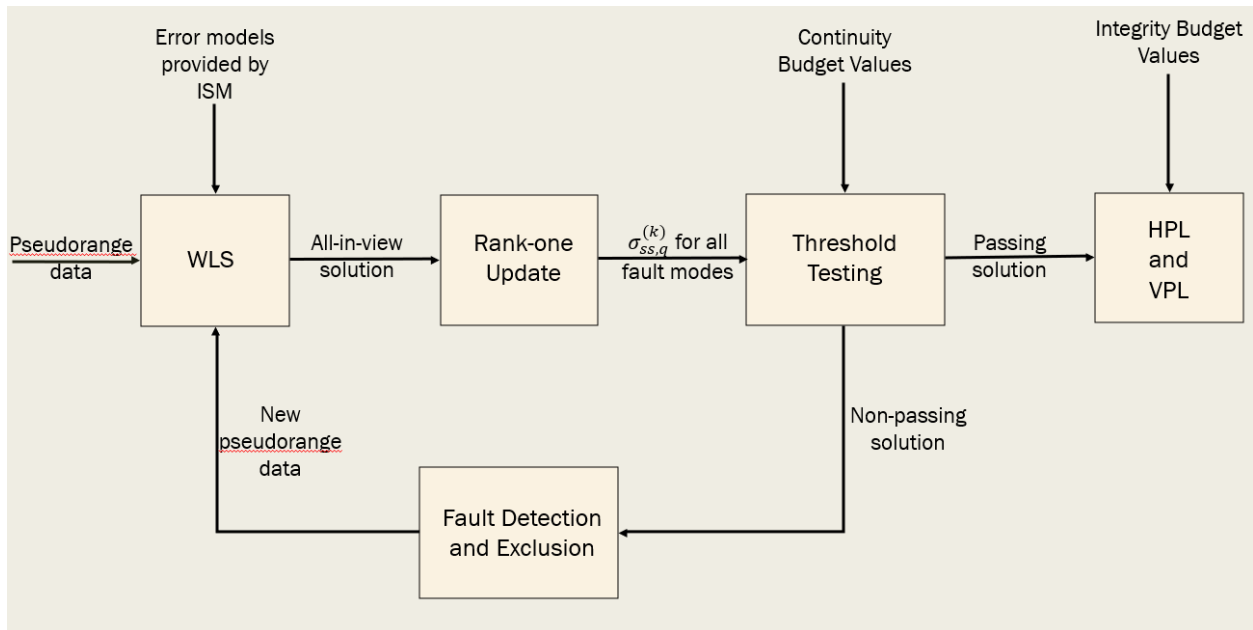


Figure 4.1: Block diagram showing how the ARAIM algorithm runs through each step.

The solution, its residuals, and the corresponding covariance are formed through the use of least squares calculations. These calculations use the geometry matrix, pseudorange measurements, and a weighting matrix. The weighting matrix used is formed using the pseudorange error diagonal covariance matrix. It takes into account the covariance matrices of user error, the standard deviation

of the clock and ephemeris error, and the tropospheric delay as shown below [1]. The standard deviations of clock and ephemeris error are provided by the ISM and thus are unique to each flight.

$$C_{int}(i, i) = \sigma_{URA,i}^2 + \sigma_{tropo,i}^2 + \sigma_{user,i}^2 \quad (10)$$

$$C_{acc}(i, i) = \sigma_{URE,i}^2 + \sigma_{tropo,i}^2 + \sigma_{user,i}^2 \quad (11)$$

The inverse of this covariance matrix is taken to calculate the weighting matrix. This, the geometry matrix, and the difference in pseudorange may then be used to form the residuals through a weighted least squares estimation [1].

$$\Delta x = (G^T W G)^{-1} G^T W \Delta \rho \quad (12)$$

The residuals formed through this particular equation with the initial matrices provided is deemed the all-in-view solution. This is the solution in which all satellites seen are included and if deemed to not be faulted, satisfies the integrity budget and thus may be trusted. The all-in-view solution is also used as a basis for forming the solution for each considered fault mode.

Now that the ‘main solution’ is formed in the all-in-view solution, the solutions may be formed that represent the possible fault modes. These solutions are formed systematically by removing one satellite at a time and re-performing the least-squares estimations. This is made easier by performing a convenient rank-one update that is the difference between two position solutions, the all-in-view and the solution for the removed satellite. These may be calculated as shown in Appendix I of [1] and the equations for both position update and covariance update shown below.

$$x - x_i = \frac{(G^T W G)^{-1} g_i w_i}{1 - g_i^T w_i (G^T W G)^{-1} g_i} (y_i - g_i (G^T W G)^{-1} G^T W y) \quad (13)$$

$$(G_i^T W_i G_i)^{-1} = (G^T W G)^{-1} + \frac{(G^T W G)^{-1} (g_i w_i g_i^T) (G^T W G)^{-1}}{1 - g_i^T w_i (G^T W G)^{-1} g_i} \quad (14)$$

In the above equation, the lower-case \mathbf{g} and \mathbf{w} represent the portions of data that are produced by the i^{th} satellite. The covariance needed to be calculated as it is used in the calculation of \mathbf{S} , which is the difference in satellite's position from the previous time step and calculated solely in terms of satellite geometry and ultimately multiplied by the difference in pseudorange to form the residuals. In the following equation, \mathbf{k} represents the fault mode being considered [1].

$$\mathbf{S}^{(k)} = (G^T W^{(k)} G)^{-1} G^T W^{(k)} \quad (15)$$

The result of this calculation is then used to calculate the variance of the difference in residuals between the all-in-view and fault mode solutions as shown. In the following equation, the superscript $\mathbf{0}$ represents the all-in-view solution, the subscript \mathbf{q} represents the direction being considered, East, North, or Up, and \mathbf{e}_q represents a vector with the q^{th} value being one and all others zero [1].

$$\sigma_{SS,q}^{(k)2} = \mathbf{e}_q^T (\mathbf{S}^{(k)} - \mathbf{S}^{(0)}) \mathbf{C}_{acc} (\mathbf{S}^{(k)} - \mathbf{S}^{(0)})^T \mathbf{e}_q \quad (16)$$

Once the variance for the respective distance between the all-in-view and each fault mode solution is calculated, the thresholds may be calculated and their corresponding test performed. The thresholds and tests are calculated for all three directions and each have different coefficients that must be calculated, however, East and North have the same coefficient. They are calculated below with \mathbf{Q} representing the quantile of a zero-mean unit-variance Gaussian distribution, \mathbf{N}_{fault} as the number of fault modes possible, and \mathbf{P} as the integrity budget allowed for the horizontal and vertical directions [1]. The integrity budget values used in these equations are pre-set values and may be adjusted to test or potentially improve performance in failure detection, for this thesis work the values provided in [1] were used.

$$K_{fa,1} = K_{fa,2} = Q^{-1}\left(\frac{P_{FA_HOR}}{4N_{fault}}\right) \quad (17)$$

$$K_{fa,3} = Q^{-1}\left(\frac{P_{FA_VERT}}{2N_{fault}}\right) \quad (18)$$

The threshold may then be calculated, represented by T , and the test performed using the following [1]:

$$T_{k,q} = K_{fa,q} \sigma_{ss,q}^{(k)} \quad (19)$$

$$\tau_{k,q} = \frac{|x_q^{(k)} - x_q^{(0)}|}{T_{k,q}} \leq 1 \quad (20)$$

If the above test shown in equation 20 is not passed, then exclusion will have to be performed on the data as a fault is present. A fault may be present in one of two forms: satellite in which a single satellite is faulted, or constellation in which a full constellation of satellites is deemed to be faulted.

If the above test is passed, however, the horizontal protection limit, vertical protection limit, and other performance parameters for the position solution may be calculated. As with calculating the test thresholds, the protection limits are defined per each direction. The horizontal protection limit is calculated in the north/south and east/west directions while the vertical protection limit is calculated in the vertical directions, up/down. The horizontal and vertical protection limits may be calculated by solving the following equations for the variables HPL and VPL [1].

$$\begin{aligned}
& 2Q\left(\frac{VPL - b_3^{(0)}}{\sigma_3^{(0)}}\right) + \sum_{k=1}^{N_{fault\ modes}} p_{fault,k} Q\left(\frac{VPL - T_{k,3} - b_3^{(k)}}{\sigma_3^{(k)}}\right) \\
& = PHMI_{VERT} \left(1 \right. \\
& \quad \left. - \frac{P_{sat,not\ monitored} + P_{const,not\ monitored}}{PHMI_{VERT} + PHMI_{HOR}} \right)
\end{aligned} \tag{21}$$

$$\begin{aligned}
& 2Q\left(\frac{HPL_q - b_q^{(0)}}{\sigma_q^{(0)}}\right) + \sum_{k=1}^{N_{fault\ modes}} p_{fault,k} Q\left(\frac{HPL_q - T_{k,q} - b_q^{(k)}}{\sigma_q^{(k)}}\right) \\
& = \frac{1}{2} PHMI_{HOR} \left(1 \right. \\
& \quad \left. - \frac{P_{sat,not\ monitored} + P_{const,not\ monitored}}{PHMI_{VERT} + PHMI_{HOR}} \right)
\end{aligned} \tag{22}$$

In the above equations, T is the test threshold, σ represents the variance calculated in the given direction, $P_{sat, not\ monitored}$ and $P_{const, not\ monitored}$ represents the probability that a satellite or constellation is not being monitored properly at that epoch, $PHMI_{VERT}$ and $PHMI_{HOR}$ are the integrity budgets in the vertical and horizontal as provided in the integrity budget, Q is once again the quantile of a zero-mean unit-variance Gaussian distribution, b is the bias represents the worst case impact on the position solution, and p_{fault} is the probability of a fault occurring. The subscripts and superscripts, k, q, θ , used in the previous two equations represent current fault mode, current direction, and all-in-view solution respectfully. The bias and probability of fault variables shown may be calculated by the following equations [1]:

$$b_q^{(k)} = \sum_{i=1}^{N_{sat}} |S_{q,i}^{(k)}| b_{nom,i} \quad (23)$$

$$p_{fault,k} = \prod_{s=1,\dots,r} P_{event,i_s} \quad (24)$$

In the above equations $b_{nom,i}$ represents the maximum bias that may be present on the nominal for a particular satellite i . The subscript s in equation 24 represents the fault mode being tested at that particular instance.

The horizontal protection limit requires an additional calculation to be fully formed. Since it takes into account both the measurements from the north/south and east/west directions, it must be calculated as follows [1]:

$$HPL = \sqrt{HPL_1^2 + HPL_2^2} \quad (25)$$

In addition to these protection levels, other performance criteria may be found with the use of the solution produced. The additional values calculated within this algorithm are the ninety-five-percentile accuracy bound, the fault-free position error bound, and the effective monitor threshold (EMT) as calculated below [1].

$$accuracy(95\%) = K_{ACC} \sigma_{v,acc} \quad (26)$$

$$fault\ free\ (10^{-7}) = K_{FF} \sigma_{v,acc} \quad (27)$$

$$EMT = \max(T_{k,3}) \quad (28)$$

In the previous two equations, the \mathbf{K} variables are pre-set values that may be adjusted prior to running the algorithm. The term $\sigma_{v,acc}$ is the standard of deviation of the vertical position solution and may be calculated in the following equation with previously defined terms [1].

$$\sigma_{v,acc} = \sqrt{e_3^T S^{(0)} C_{acc} S^{(0)T} e_3} \quad (29)$$

4.2 Fault Detection

4.2.1 Minimum Chi-Squared Statistic

Once a fault mode formed from the all-in-view solution does not pass the aforementioned threshold test, a fault is deemed to be present. In order to detect this fault, the minimum chi-squared test statistic must be determined from the available fault modes. The chi-squared test statistic is chosen because it coincides with the satellite that produces the position solution that is furthest from the all-in-view solution as stated in Appendix F of [1]. The relation between the maximum normalized solution separation and the minimum chi squared test statistic is shown below as reproduced from [1].

$$\chi^2 = \chi_i^2 + \left(\frac{x_q^i - x_q^{(0)}}{\sigma_{ss,q}^i} \right)^2 \quad (30)$$

In the above equation, the χ_i term represents the chi squared statistic as calculated with the particular satellite i left out or excluded. Since this is an effective strategy for single satellite faults, it then follows in this algorithm that multiple failed satellites may be detected by running the dataset through the testing and fault detection then exclusion steps multiple times until the test for the all-in-view solution is passed.

4.2.2 Rank-one Updates

The chi square statistic for each fault mode may be calculated in the same method as the position solution and covariance matrix updates, through a rank-one update. This may be calculated as shown in Appendix I of [1].

$$\chi_i^2 = \chi^2 - \frac{w_i}{1 - g_i^T w_i (G^T W G)^{-1} g_i} (y_i - g_i^T (G^T W G)^{-1} G^T W y)^2 \quad (31)$$

In the above equation, the lower-case \mathbf{g} and \mathbf{w} represent the portions of data that are produced by the i^{th} satellite. The results of this calculation produce the chi square statistic for the all-in-view data with the contribution of the suspected failed satellite subtracted out.

4.3 Fault Exclusion

Once the failed satellite is identified all of its data must be excluded from the variables that are used to calculate the position solution, and the all-in-view solution calculated again. To exclude the selected satellite's data, a very simple method was utilized. A method was written within the algorithm that used the stored index for the failed satellite to remove all of the satellite's data from the necessary variables. It is very important in the building of this new data to ensure that the correct satellite's data is deleted and that it is deleted from all of the data used in subsequent steps to rerun the integrity tests on the all-in-view solution. To ensure that this occurred, the index for the faulted

satellite and the sizes of the variables before and after exclusion were displayed within the printed statement during testing.

Chapter 5 Testing

All data used in this testing was created through the Matlab SatNav Toolbox [34]. The simulation produced data for a flight along the trajectory shown within the plot below. Within the Python code that ran the algorithm, a method for injecting error to the data was added. This method worked by generating an uniformly distributed random number, and if that number fell beneath the set probability, for these tests ten percent, a ten meter error would be added to the smoothed range data of a random satellite for the current epoch. If error is added to one satellite, then the same process would be done for a second satellite in that same epoch in which there will once again be a ten percent chance of an error being injected. Further testing was done as well with errors of five, fifty, and one-hundred meters being injected in the same random chance method. This additional testing was done to explore the capability of the fault detection and exclusion methods with various sizes of error.

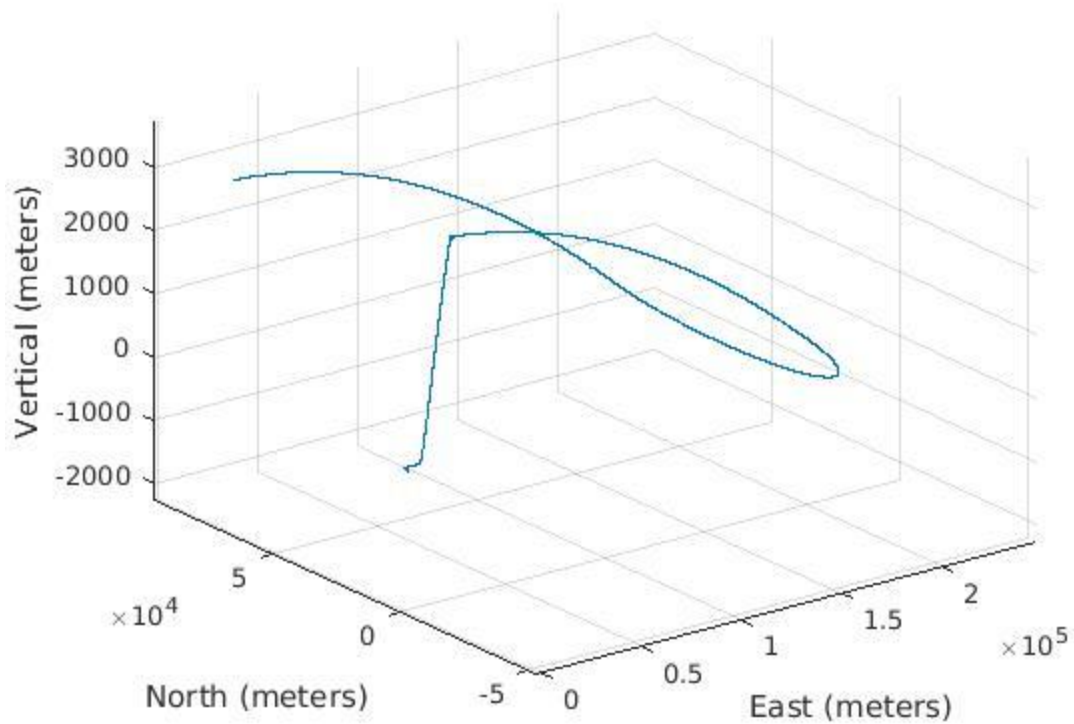


Figure 5.1: Trajectory of simulated flight.

5.1 Sensitivity Analysis

Since it is an integrity algorithm that is based off of a weight least squares (WLS) estimate, it was logical to compare the positioning results computed from both ARAIM and WLS. There were, however, two forms of the WLS solution formed in order to further check the effectiveness of the ARAIM method. The first of these two created solutions was WLS in which the failed satellites were known and excluded automatically from the solution. The second was WLS with no excluded satellites at all. Five runs over the full duration of the test of the data set, each with unique random injected errors, were made and the position solutions of ARAIM, both WLS methods, and the known truth were compared.

5.1.1 Chi-Square Statistic

Since the method for detecting the faulted pseudorange observation is the minimum chi-squared test statistic, the demonstration of this being proved to be true must be made. To accomplish this, a short run of the ARAIM algorithm in which error is added to the data of a satellite of a selected epoch was performed. The calculated chi-square statistic of each fault mode and the index of the removed satellite was calculated and compared. The performance of these methods was measured against two additional factors as well. In order to measure the validity of the ten-meter error applied to the pseudorange measurements as a standard, errors larger and smaller than ten meters were applied in separate tests. They were applied in the same manner as the ten-meter errors and used for full test runs through the entirety of the simulated data. The data taken from these test runs were then measured to observe how the effectiveness of the fault detection and exclusion methods varied with the size of the error applied.

5.1.2 Increasing Potential Fault Modes

Within each epoch each satellite must be removed for both constellations present and the algorithm worked out to obtain an integrity test result. This must be done not only for one satellite being removed in consideration of fault, but two as well. The amount of calculations increases exponentially due to these increasing values and thus should be taken into account when considering the processing time of this algorithm. This is derived based on the number of solution combinations when taking in account the number of satellites tracked, constellations used, and maximum simultaneous faults assumed. In the following equations, the variable N_{events} represents the number of events considered in relation to the number of satellites and constellations available, N_{sat} represents the number of satellites, N_{const} represents the number of constellations, $N_{faultmax}$ represents the number of simultaneous fault modes considered, and N_{modes} represents the number of fault modes to be considered.

$$N_{modes} = \frac{N_{events}!}{(N_{events} - N_{faultmax})! N_{faultmax}!} \quad (32)$$

$$N_{events} = N_{sat} + N_{const} \quad (33)$$

Chapter 6 Results

In order to perform a full analysis of the performance and capabilities of this state-of-the-art ARAIM algorithm, a few different results were looked at and comparisons made to other algorithms. The final user position in each epoch was compared between ARAIM, WLS while excluding known failed satellites, WLS with no corrections applied, and the provided truth. The difference in position solution between ARAIM and the provided truth was also used to measure how well the performance of ARAIM was to remaining within the horizontal and vertical protection limits.

The performance of ARAIM overall was measured through three different ways. First, the ability of the added fault detection and exclusion to successfully detect failures and correct for them was observed. Next, the performance of the overall ARAIM algorithm will be observed through the comparison of measured position performance values to the requirements for them as set in the International Civil Aviation Organization's (ICAO) standards and recommended practices (SARPs). The final measure of ARAIM's performance will be in an analysis of the computational load due to the large amount of computations that must be made at each epoch.

6.1 Position Error Comparison

As a way to measure the positioning capability of ARAIM as well as the ability to remain within calculated horizontal and vertical protection limits, the results of the user position calculated by ARAIM was differenced against the provided truth. The difference between the two position

solutions in all three directions, east, north, and vertical, are shown in Figures 6.1, 6.2, 6.3. It can be observed from the figures that the difference in the position solution between ARAIM and the simulation truth are well below the calculated protection levels at all times. This signifies that the position solution may be trusted due to its adherence to the protection levels set with the values from the ISM.

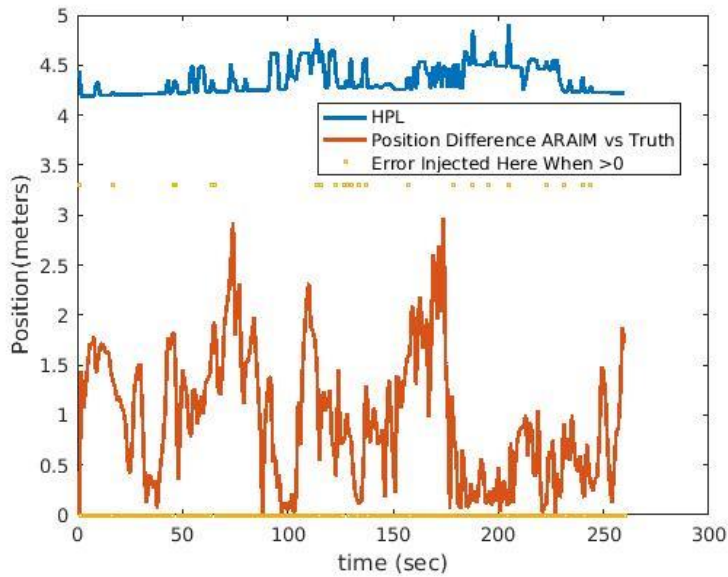


Figure 6.1: Position error for ARAIM in the east direction with HPL and markers for injected error.

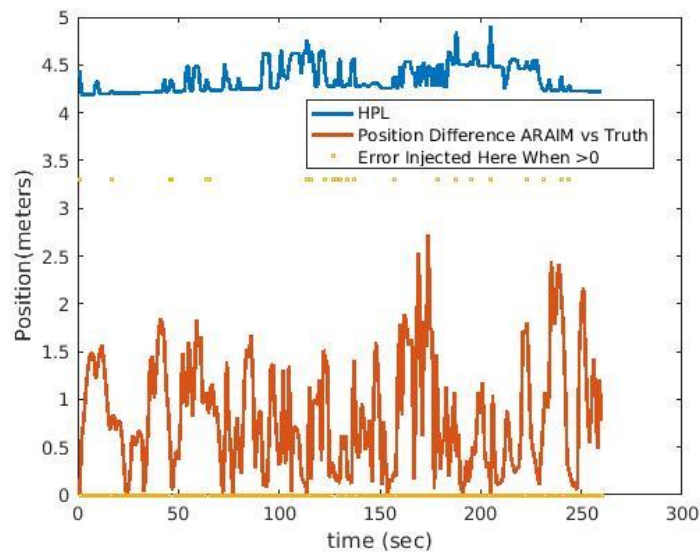


Figure 6.2: Position error for ARAIM in the north direction with HPL and markers for injected error.

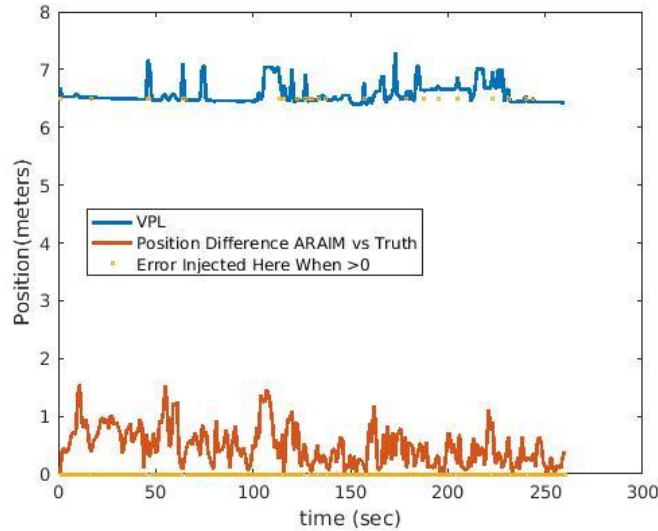


Figure 6.3: Position error for ARAIM in the vertical direction with VPL and markers for injected error.

Two observations may be made from the previous three figures. The first of which is that the position difference is well below the calculated protection levels. This was continually seen in all five tests through the entirety of the data set as seen in the following tables. The five tests were all through the entirety of the simulated data set with ten-meter error placed through the random chance method described previously.

Table 6.1: Minimum and maximum HPL and VPL and maximum position difference between ARAIM and truth.

	HPL_{max} (m)	HPL_{min} (m)	VPL_{max} (m)	VPL_{min} (m)	maxΔx_{ARAIM,Truth} (m)	maxΔy_{ARAIM,Truth} (m)	maxΔz_{ARAIM,Truth} (m)
TR1	4.91	4.19	7.29	6.39	2.72	2.97	1.55
TR2	4.93	4.19	7.30	6.39	2.72	2.97	1.66
TR3	4.75	4.19	7.05	6.39	2.58	2.92	1.53
TR4	4.92	4.19	7.62	6.39	2.72	3.09	2.09
TR5	4.66	4.19	8.22	6.39	2.72	2.97	1.55

Table 6.2: Root mean square and mean of the difference in positioning data between ARAIM and the truth.

	Δx RMS (m)	Δy RMS (m)	Δz RMS (m)	Δx mean (m)	Δy mean (m)	Δz mean (m)
TR1	1.02	1.17	0.59	0.83	0.97	0.48
TR2	1.02	1.16	0.61	0.82	0.97	0.49
TR3	0.999	1.16	0.58	0.80	0.97	0.48
TR4	1.02	1.19	0.60	0.82	0.98	0.49
TR5	1.01	1.17	0.59	0.82	0.98	0.49

Throughout all five runs the maximum difference in position between ARAIM and truth are well below the minimum protection levels calculated in all directions. This is also true for the calculated root mean square error and mean of the position solution in all three directions. The larger maximum values are due to the presence of errors, some of which are added upon by the noise present in the data. It can be seen in Figures 6.1, 6.2, and 6.3, however, that a majority of the spikes in the protection levels occur at the presence of errors and then fall back down to a lower value due to the exclusion of said errored satellites.

The position solution produced through ARAIM was also compared to the position solution produced through the use of WLS with the errored satellites already known and excluded prior to processing. The failed satellites were guaranteed to be known for WLS as a list of satellites with errors injected into their carrier smoothed pseudorange data from the ARAIM test was provided. The satellites listed were excluded from the data used in the WLS solution at the indicated epochs where an error was injected within the ARAIM method. This provides a solution against which the ARAIM solution may be compared and when there is no additional sources of error, there will be no difference between the two solutions.

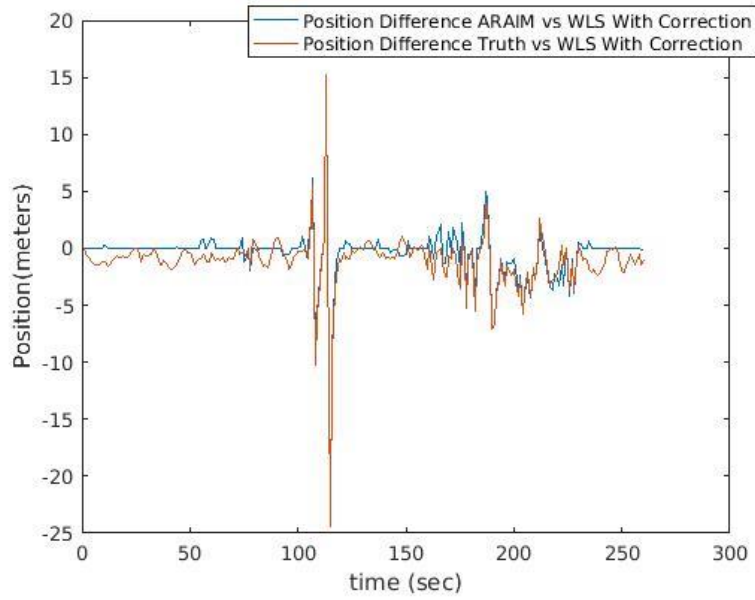


Figure 6.4: Overlay of the difference in positioning of ARAIM and WLS and difference in positioning of simulation truth and WLS in the X or east direction.

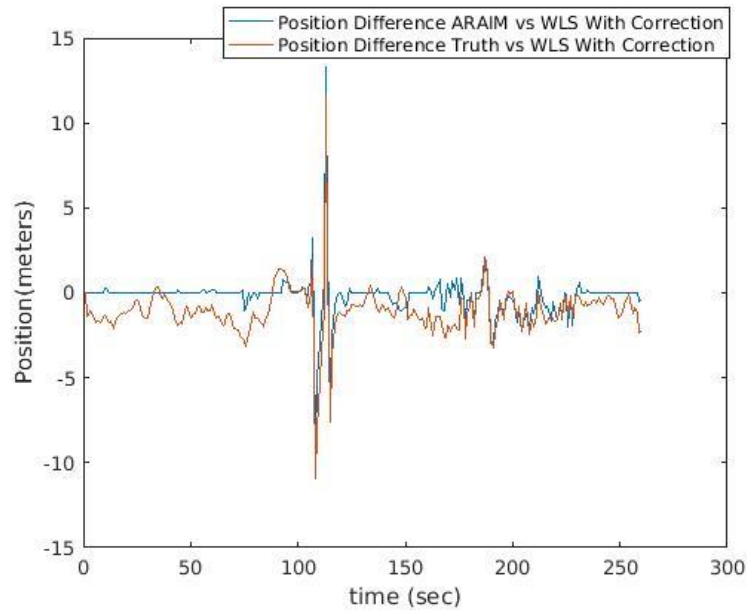


Figure 6.5: Overlay of the difference in positioning of ARAIM and WLS and difference in positioning of simulation truth and WLS in the y or north direction.

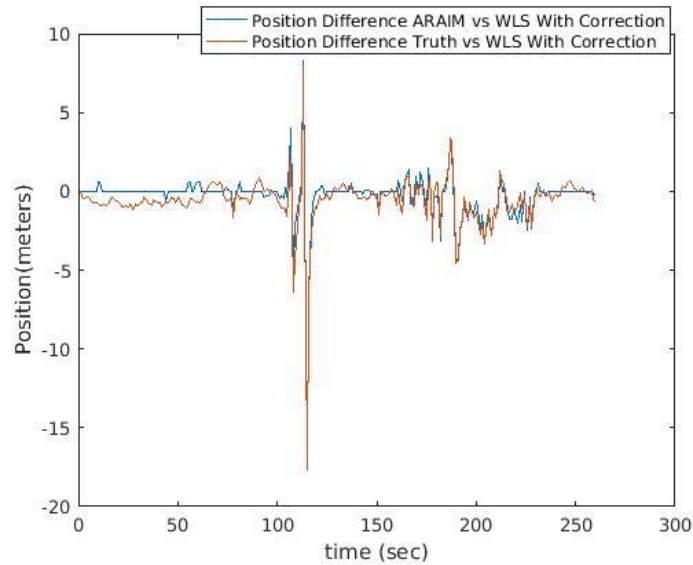


Figure 6.6: Overlay of the difference in positioning of ARAIM and WLS and difference in positioning of simulation truth and WLS in the z or vertical direction.

When comparing the difference in positioning of ARAIM and WLS with that of the provided truth and WLS it can be seen that the present difference is small. Aside from the difference caused by the noise present as seen in the position difference between ARAIM and the truth, it can be observed that the same larger increases in error are present in this comparison. The large discrepancies in position solution between ARAIM and WLS are caused by forms of error, such as multipath, which is the presence of delayed reflected signals after the arrival of the true signal, that were not injected by the code running the positioning algorithms. An additional source of error will be discussed in the fault detection and exclusion results section. ARAIM was able to detect these errors and correct the position solution because the provided satellite data produced values that did not pass threshold tests when processed in ARAIM.

6.2 Fault Detection and Exclusion Capability

Measuring the effectiveness of ARAIM goes beyond the ability of the algorithm to correct for differences in the position solutions. The ability to correct the position solution comes as a result of the fault detection and exclusion methods working as intended. To test this, a ten-meter error was

placed on a known satellite in a known epoch. The chi-squared statistic was calculated for iterations of the data set that were formed by removing individual satellites, the formed fault modes, and compared.

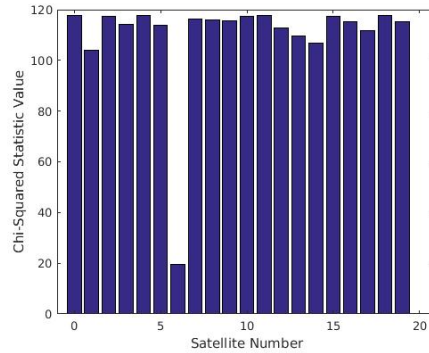


Figure 6.7: Chi-Squared Statistic Values calculated for each fault mode in one epoch before exclusion of the failed satellite.

In Figure 6.7 it can be seen that the failed satellite, satellite six, has a chi-squared statistic that is much lower than that of the others. This proves not only that the satellite whose position is furthest from the all-in-view is identified through having the lowest chi-squared statistic, but also that the ARAIM algorithm indeed works as it should. Proof of exclusion being performed on failed satellites may also be seen by looking at the difference in position solution between ARAIM and WLS with knowledge of failed satellites and WLS with knowledge of failed satellites and WLS with no corrections.

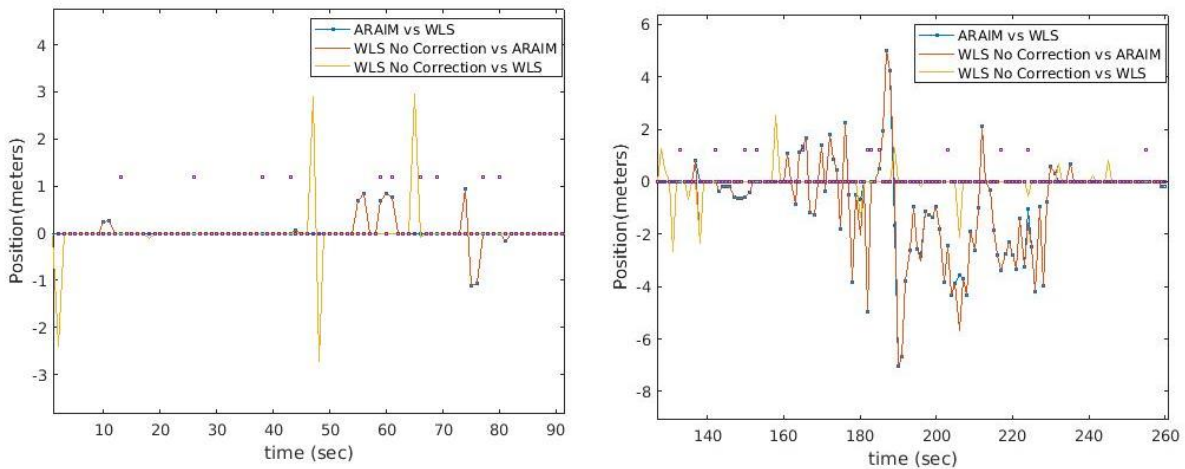


Figure 6.8: Graphs depicting an overlaying of the difference in position solution between ARAIM and WLS with corrections, ARAIM and WLS with no corrections, and WLS with corrections and without. The graphs are magnified to focus on the difference between the three position solutions and represent two spans of time within the same test period. The purple points on the graph show where there is an injected failure.

The effectiveness of the exclusion of failed satellites may be observed as the excluded satellite provides an improvement in the position solution as shown in the comparison between the difference of ARAIM and WLS with corrections and ARAIM and WLS without corrections. The improvement is equal to the difference in position solution between WLS with corrections and WLS without corrections. These improvements effect the position solution in all three different directions differently, however, as the error was originally placed upon the smoothed pseudorange.

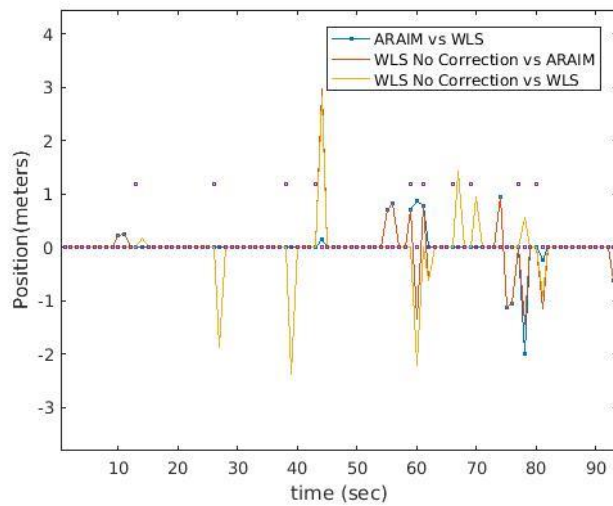


Figure 6.9: Graph depicting an overlaying of the difference in position solution between ARAIM and WLS with corrections, ARAIM and WLS with no corrections, and WLS with corrections and without in the east. The graph is magnified to focus on the difference between the three position solutions and represent a span of time within the larger test period. The purple points on the graph show where there are errors injected.

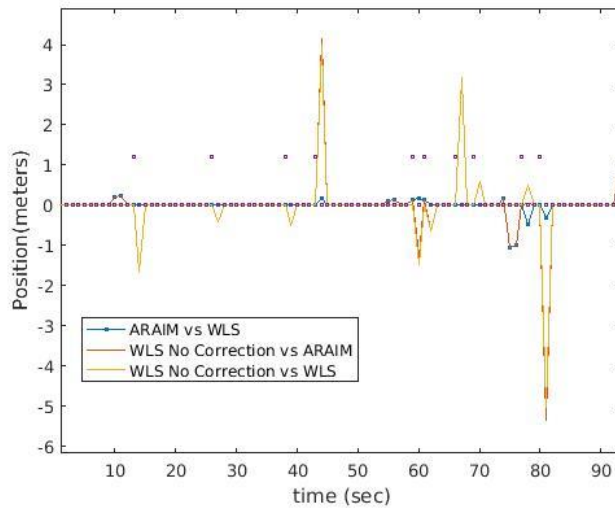


Figure 6.10: Graph depicting an overlaying of the difference in position solution between ARAIM and WLS with corrections, ARAIM and WLS with no corrections, and WLS with corrections and without in the north. The graph is magnified to focus on the difference between the three position solutions and represent a span of time within the larger test period. The purple points on the graph show where there are errors injected.

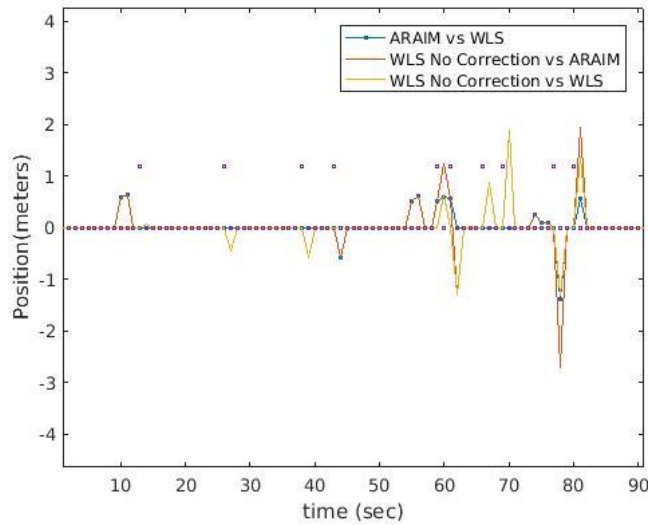


Figure 6.11: Graph depicting an overlaying of the difference in position solution between ARAIM and WLS with corrections, ARAIM and WLS with no corrections, and WLS with corrections and without in the vertical. The graph is magnified to focus on the difference between the three position solutions and represent a span of time within the larger test period. The purple points on the graph show where there are errors injected.

The added fault detection and exclusion methods were also to be measured for performance in regards to potential false exclusions and missed satellite failures. A potential false exclusion includes failures that are detected that are not attributed to the failures injected upon the data. The results for a varied range of errors, five, ten, fifty, and one-hundred meters, were tabulated.

Table 6.3: Results of fault exclusion by varying injected fault size.

Fault Size (m)	Correctly Identified Faults	False Exclusions	Missed Faults
5	16	136	8
10	23	143	0
50	34	141	0
100	25	145	0

The missed faults are a direct cause of the size of the error injected upon the pseudorange data. When it is below the chosen size of ten meters the injected failures are oftentimes missed while when set at or above ten meters they are always detected. This occurs because the error caused by noise and causes of error outside of the injected faults are typically around ten meters. An additional

cause of these errors outside of the previously mentioned sources was observed to be poor geometry of particular satellites. This was seen when looking at the failures produced and the satellite that was found to have the lowest chi-squared statistic when the threshold testing was not passed. It was observed that when a false exclusion was made, it was often made on the same satellite over multiple epochs in a row. It was also seen to be repeatable as the same false exclusions were seen on the satellite at the same epochs over all tests regardless of the size of the injected error. Such a behavior in the quality of the geometry is seen when a satellite is moving towards the horizon and out of the sight of the GPS receiver. This behavior is also linked to the large spikes in error when looking at the position difference between the weighted least squares solution with corrections and the truth. Since it is only excluding the satellites that are known to have failures injected upon them, the weighted least-squares method cannot correct for this error source.

6.3 Position Performance Measurements

Since this is an algorithm in which the integrity of a positioning solution is being observed, solutions formed by it should abide to the ICAO SARPs [2]. Performance results remaining below these standards signifies that the algorithm may be deemed acceptable for civil aviation. The maximum and minimum of these calculated values for each run of the entire data set were recorded and compared to the standards set in [2].

Table 6.4: ICAO SARPs

Performance Criteria	Value (m)
95% Accuracy	4
99.99999% Fault-free Accuracy	10
Effective Monitor Threshold	15
Limit on the Position Error	35

Table 6.5: Performance of ARAIM in each test run for ten-meter injected errors

	Acc_{95%,max}	Acc_{95%,min}	PosErr_{FF,max}	PosErr_{FF,min}	EMT_{max}	EMT_{min}
TR1	1.44	1.23	3.92	3.34	3.39	2.65
TR2	1.44	1.23	3.92	3.34	3.39	2.65

TR3	1.38	1.23	3.76	3.34	3.39	2.65
TR4	1.45	1.23	3.95	3.34	3.64	2.65
TR5	1.56	1.23	4.24	3.34	3.39	2.65

The performance of ARAIM remains well under the set performance criteria standards as seen in the previous two tables. This is also observed in the position results as they remain well under the limit on the position error of thirty-five meters.

6.4 Increasing Potential Fault Modes

As this algorithm is being performed on every epoch of data and must be re-performed to a point, it is rather computationally dense. This becomes moreso as within each epoch a fault mode, in which each satellite from both the GPS and GALILEO constellations seen are to be removed from the solution both individually and in pairs, is formed and threshold testing performed upon it. This increases exponentially as more and more faults are considered to be occurring simultaneously. It can be seen from the graphs below that increasing the number of simultaneous faults that are considered to be present causes the number of fault modes that must be formed and tested to increase exponentially. These increases then in turn cause an increase in calculation time, adversely affecting the efficiency of the algorithm.

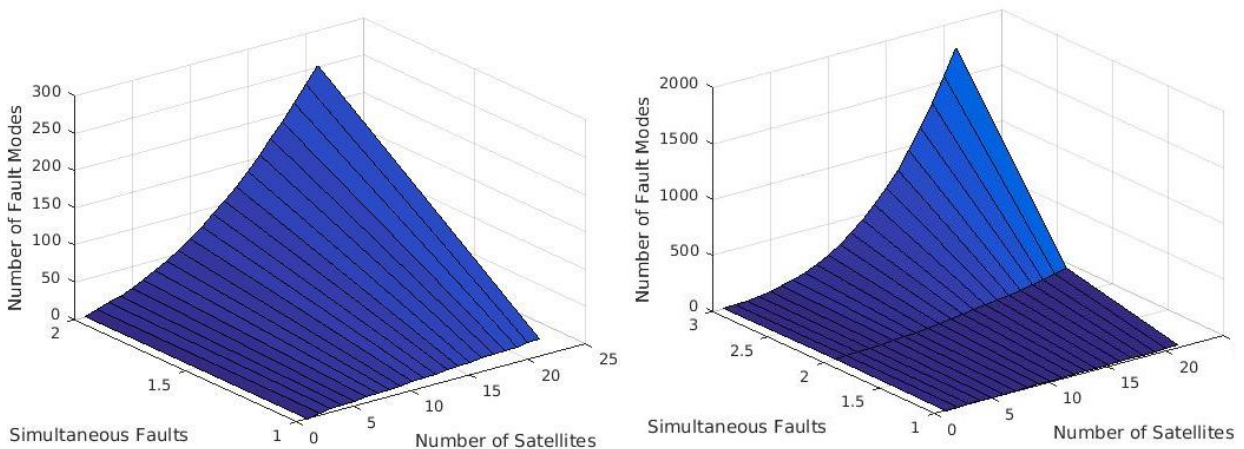


Figure 6.12: The amount of fault modes needed for consideration of increasing number of possible simultaneous faults from 2 to 3. The graph on the left shows the increase from 1 to 2 simultaneous faults and the graph on the right shows the increase from 1 to 3 simultaneous faults.

Chapter 7 Conclusions and Future Work

7.1 Conclusions

A state-of-the-art solution separation ARAIM algorithm was implemented as it is written in [1]. The basis of this thesis work in particular was to add a fault detection and exclusion method into an existing RAIM script and fully implement the ARAIM algorithm. This was successfully accomplished and the performance of the algorithm compared to the position solution produced through a normal WLS algorithm. The results of this comparison showed that ARAIM produced a better position solution with less error when compared to the provided truth position. Specifically, ARAIM was able to mitigate multipath and geometry errors that were seen to adversely affect the WLS position solution. The performance of the ARAIM algorithm's positioning was further measured against calculated protection levels and found to be well within them as well as the ICAO SARPs. The capability of the ARAIM algorithm's fault detection and exclusion methods were also found to be effective as a direct testing of them proved that a satellite with a ten-meter fault injected upon the carrier smoothed pseudorange data can be successfully detected and excluded. The performance of these methods was measured further against faults that were larger and smaller than ten-meters. It was seen through these tests that faults greater than ten-meters were always detected and excluded, but faults less than ten-meters were at times missed. Although a multitude of false exclusions were made, these were caused by errors produced through changes in geometry and thus the capability of ARAIM was proven further. The effects of the exclusions made were examined through juxtaposition of the difference in position between ARAIM and the truth, ARAIM and WLS with corrections, and WLS with corrections and WLS with no corrections on the same graph.

In addition, the amount of solutions necessary to be formed was calculated and it was found that the value increased exponentially as constellations and satellites are added and more simultaneous fault modes are considered. This exponential increase in fault modes to be considered and calculated for slows down the algorithm and thus a method for reducing this calculation time is needed.

7.2 Future Work

The study into this ARAIM algorithm is not complete, as a few additional things need to be investigated. Assumed to be correct when choosing to exclude a satellite, this algorithm will need to have the ability to consider the legitimacy of its decision to exclude a satellite's data for each epoch. There are also improvements suggested within [1] that should be considered for this algorithm as well as improvements to the large computation time as suggested in [29].

This algorithm will be used in an investigation into the development of a robust GNSS processing techniques as a basis on which to be compared to [8] [9]. It will also serve as a source for any researcher at West Virginia University who wishes to know how solution separation ARAIM is to operate for aviation.

References

- [1] J. Blanch, T. Walker, P. Enge, Y. Lee, B. Pervan, M. Rippl, A. Spletter and V. Kropp, "Baseline Advanced RAIM User Algorithm and Possible Improvements," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 1, pp. 713-730, 2015.
- [2] International Civil Aviation Organization (ICAO), "Annex 10: Aeronautical Communications," *Volume 1: Radio Navigation Aids - Amendment 84*, 2009.
- [3] E. D. Kaplan and C. J. Hegarty, *Understanding GPS Principles and Applications*, Norwood: Artech House Incorporated, 2006.
- [4] R. Connor, "Twenty Years of GPS and Instrument Flight," *Smithsonian Air and Space Museum Aeronautics Department*, 24 February 2014.
- [5] Department of Defense, Department of Homeland Security, and Department of Transportation, "2008 Federal Radionavigation Plan," 2008.
- [6] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*, Lincoln: Ganga-Jamuna Press, 2012.
- [7] J. Blanch, T. Walter, P. Enge, S. Wallner, F. A. Fernandez, R. Dellago, R. Ioannides, I. F. Hernandez, B. Belabbas, A. Spletter and M. Rippl, "Critical Elements for a Multi-Constellation Advanced RAIM," *Navigation: Journal of the Institute of Navigation*, vol. 60, no. 1, pp. 53-69, 2013.
- [8] R. Watson and J. Gross, "Evaluation of Kinematic Precise Point Positioning Convergence with an Incremental Graph Optimizer," in *IEEE/ION Position, Location, and Navigation Symposium (PLANS)*, Monterey, 2018.
- [9] R. Watson and J. Gross, "Robust Navigation in GNSS Degraded Environment Using Graph Optimization," in *Proceedings of the 30th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2017)*, Portland, 2016.
- [10] C. J. Hegarty, "Appendix A: Least Squares and Weighted Least Squares Estimates," in *Understanding GPS Principles and Applications*, Norwood, Artech House Incorporated, 2006, pp. 663-664.
- [11] R. G. Brown, "Receiver Autonomous Integrity Monitoring," *Progress in Astronautics and Aeronautics: Global Positioning System: Volume II: Theory and Applications*, pp. 143-165, 1993.
- [12] Y. C. Lee, "Analysis of Range and Position Comparison Methods as a Means to Provide GPS Integrity in the User Receiver," *Proceedings of The Institute of Navigation*, pp. 5-19, 1986.
- [13] B. W. Parkinson and P. Axelrad, "Autonomous GPS Integrity Monitoring Using the Pseudorange Residual," *Navigation: Journal of The Institute of Navigation*, pp. 255-274, 1988.

- [14] M. A. Sturza, "Navigation System Integrity Monitoring Using Redundant Measurements," *Navigation: Journal of The Institute of Navigation*, pp. 483-501, 1989.
- [15] R. G. Brown, "A Baseline GPS RAIM Scheme and a Note on the Equivalence of Three RAIM Methods," *Navigation: Journal of The Institute of Navigation*, pp. 101-116, 1992.
- [16] R. G. Brown and G. Y. Chin, "GPS RAIM: Calculation of Threshold and Protection Radius Using Chi-Square Methods - A Geometric Approach," *Global Positioning System: GPS and Its Augmentation Systems*, vol. 5, pp. 155-178, 2002.
- [17] B. Korn, H.-U. Doehler and P. Hecker, "Navigation Integrity Monitoring and Obstacle Avoidance Detection for Enhanced-Vision Systems," in *Aerospace/Defense Sensing, Simulation, and Controls*, Orlando, 2001.
- [18] A. Fakharian, T. Gustafsson and M. Mehrfam, "Adaptive Kalman Filtering Based Navigation; An IMU/GPS Integration Approach," in *2011 International Conference on Networking, Sensing, and Control*, Delft, 2011.
- [19] R. Da and C.-F. Lin, "A New Failure Detection Approach and its Application to GPS Autonomous Integrity Monitoring," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 31, no. 1, pp. 499-506, 1995.
- [20] Y.-C. Chao, "Real Time Implementation of the Wide Area Augmentation System for the Global Positioning System With and Emphasis on Ionospheric Modeling," *Dissertation Abstracts International*, vol. 58, no. 07, p. 3756, 1997.
- [21] P. Y. Hwang and R. G. Brown, "RAIM FDE Revisited: A New Breakthrough in Availability Performance with NIO-RAIM (Novel Integrity-Optimized RAIM)," *Navigation: Journal of the Institute of Navigation*, vol. 53, no. 1, pp. 654-665, 2005.
- [22] Y. C. Lee, "Two New RAIM Methods Based on the Optimally Weighted Average Solution (OWAS) Concept," *Navigation: Journal of the Institute of Navigation*, vol. 54, no. 4, pp. 333-345, 2007.
- [23] B. S. Pervan, S. P. Pullen and J. R. Christie, "A Multiple Hypothesis Approach to Satellite Navigation Integrity," *Navigation: Journal of the Institute of Navigation*, vol. 45, no. 1, pp. 61-71, 1998.
- [24] J. Blanch, A. Ene, T. Walter and P. Enge, "An Optimized Multiple Hypothesis RAIM Algorithm for Vertical Guidance," in *Proceedings of the ION GNSS*, 2007.
- [25] C. E. Cohen, H. S. Cobb, D. G. Lawrence, B. S. Pervan, J. D. Powell, B. W. Parkinson, G. J. Aubrey, W. Loewe, D. Ormiston, B. D. McNally, D. N. Kaufmann, V. Wullschleger and R. J. Swider JR, "Autoland a 737 Using GPS Integrity Beacons," *Journal of the Institute of Navigation*, vol. 42, no. 3, pp. 467-486, 1995.

- [26] G. Schroth, A. Ene, J. Blanch, T. Walter and P. Enge, "Failure Detection and Exclusion Via Range Consensus," in *European Navigation Conference*, Toulouse, 2008.
- [27] M. Rippl, G. Schroth, B. Belabbas and M. Meurer, "A Probabilistic Assessment on the Range Consensus (RANCO) RAIM Algorithm," in *International Technical Meeting of the Institute of Navigation*, Anaheim, 2009.
- [28] M. Joerger, F.-C. Chan and B. Pervan, "Solution Separation Versus Residual-Based RAIM," *Navigation: Journal of the Institute of Navigation*, vol. 61, no. 4, pp. 273-291, 2014.
- [29] P. Massatt and A. Binder, "An Improved Algorithm for Receiver Implementation of ARAIM," The Aerospace Corporation, Los Angeles, 2018.
- [30] P. Misra, E. Bayliss, R. Lafrey, M. Pratt and R. Muchnik, "Receiver Autonomous Integrity Monitoring (RAIM) of GPS and GLONASS," *Navigation: Journal of the Institute of Navigation*, vol. 40, no. 1, pp. 299-316, 1993.
- [31] W. Y. Ochieng, K. F. Sheridan, K. Sauer, X. Han, P. A. Cross, S. Lannelongue, N. Ammour and K. Petit, "An Assessment of the RAIM Performance of Combined Galileo/GPS Navigation System Using the Marginally Detectable Errors (MDE) Algorithm," *GPS Solutions*, vol. 5, no. 3, pp. 42-51, 2002.
- [32] T. Reid, T. Walter, J. Blanch and P. Enge, "GNSS Integrity in the Arctic," *Navigation: Journal of the Institute of Navigation*, vol. 63, no. 4, pp. 469-492, 2016.
- [33] A. J. Dorsey, W. A. Marquis, P. M. Fyfe, E. D. Kaplan and L. F. Wiederholt, "GPS System Segments," in *Understanding GPS Principles and Applications*, Norwood, Artech House, Inc., 2006, pp. 67-112.
- [34] GPSoft, "Satellite Navigation TOOLBOX 3.0 User's Guide," 2003.