

REAL-TIME ARAIM USING GPS, GLONASS, AND GALILEO

BY

RYAN CASSEL

Submitted in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical and Aerospace Engineering
in the Graduate College of the
Illinois Institute of Technology

Approved



Advisor

Chicago, Illinois
May 2017

ACKNOWLEDGMENT

I would first like to express my gratitude towards my academic and research advisor, Dr. Boris Pervan, for giving me the opportunity to conduct this research, helping me establish a foundational understanding of the necessary material, and supporting my efforts over these last two years. His willingness to answer questions and work through important topics with me has proven invaluable.

I would also like to thank my current and former colleagues in the Illinois Institute of Technology Navigation Lab, many of whom have become good friends: Dr. Mathieu Joerger, Dr. Samer Khanafseh, Dr. Mike Jamoom, Dr. Cagatay Tanil, Stefan Stevanovic, Yawei Zhai, Adriano Canolla, Roohollah Parvizi, Jaymin Patel, and Austin Gonzalez. I would especially like to thank Dr. Joerger and Yawei Zhai for their help specific to my work in ARAIM; it is by their knowledge and prior work that I have been able to accomplish the goals of this research. Additionally, I would like to acknowledge Santiago Perea Díaz of the German Aerospace Center (DLR) for sharing his knowledge of Galileo.

Next, I would like to thank my family for their support over these last two years and throughout my college career: my mother, Adrienne, my father, Kevin, my brother, Nathan, and my sister, Sarah. My father, a professor at IIT, has given me advice as both a professor and a father that has helped build my confidence and allow me to overcome obstacles in my work.

Finally, I would like to thank my fiancée, Marlee. She has shown me unconditional love in everything I do, and she has encouraged me every time I was overwhelmed. Her constant support is what has pushed me to achieving this accomplishment.

TABLE OF CONTENTS

	Page
ACKNOWLEDGMENT	iii
LIST OF TABLES	vi
LIST OF FIGURES	viii
ABSTRACT	ix
CHAPTER	
1. INTRODUCTION	1
1.1. GNSS Architecture	1
1.2. Current GNSS	2
1.3. RAIM	6
2. GNSS POSITIONING	7
2.1. Code Measurements	8
2.2. Carrier Phase Measurements	8
2.3. Measurement Errors	9
2.4. Ionosphere-Free Measurements	12
2.5. Carrier Smoothing	13
2.6. Satellite Orbit Computation	14
2.7. Position Estimation	18
3. METHODOLOGY	23
3.1. Multiple Hypothesis Solution Separation	23
3.2. Continuity Risk	24
3.3. Integrity Risk	32
4. ARAIM ALGORITHM	38
4.1. Error Model	38
4.2. Integrity Support Message	39
4.3. Protection Levels	42
4.4. Availability	53
4.5. Fault Detection	56
4.6. Fault Exclusion	58
5. RESULTS	62
5.1. GPS Versus GPS + GLONASS for H-ARAIM	63

5.2. H-ARAIM Versus V-ARAIM using GPS + GLONASS + Galileo	64
5.3. V-ARAIM with Degraded GLONASS Performance	69
6. IMPROVEMENTS	74
6.1. Satellite Selection	75
6.2. Fault Mode Grouping	86
7. CONCLUSION	92
APPENDIX	93
A. CARRIER SMOOTHING EQUATION DERIVATION	94
B. GLONASS SATELLITE ORBIT COMPUTATION	96
C. ARAIM AIRBORNE ERROR MODEL	100
D. PROTECTION LEVEL EQUATION DERIVATION	103
E. METHOD OF SOLVING PROTECTION LEVEL EQUATION	112
BIBLIOGRAPHY	115

LIST OF TABLES

Table	Page
1.1 Galileo Signals.	5
2.1 Minimum Number of Satellites Required for Estimation.	8
2.2 GNSS Measurement Error Sources.	12
2.3 GPS and Galileo Clock and Ephemeris Parameters.	16
2.4 GLONASS Clock and Ephemeris Parameters.	17
4.1 ISM Contents.	40
4.2 Fault Probabilities for H-ARAIM and V-ARAIM.	40
4.3 Number of Satellites Required for Detection for H-ARAIM.	45
4.4 Number of Satellites Required for Detection for V-ARAIM.	45
4.5 Number of Satellites Required for Exclusion for H-ARAIM.	48
4.6 Number of Satellites Required for Exclusion for V-ARAIM.	49
4.7 RNP and LPV-200 Operational Requirements.	54
5.1 Continuity and Integrity Budget Allocations for V-ARAIM and H-ARAIM.	62
5.2 Nominal ISM Parameters for H-ARAIM.	63
5.3 Nominal ISM Parameters for V-ARAIM.	65
5.4 Degraded ISM Parameters for V-ARAIM.	70
C.1 Galileo User Error.	102

LIST OF FIGURES

Figure	Page
1.1 GNSS Architecture.	2
1.2 GPS (left), Galileo (middle), and GLONASS (right) Constellations.	5
2.1 Position Estimation by Trilateration.	7
2.2 Signal-in-Space Error.	10
2.3 Ionosphere and Troposphere.	10
2.4 Multipath.	11
2.5 Carrier Smoothing Block Diagram.	14
3.1 Real-Time ARAIM Implementation Flow Chart.	23
3.2 MHSS Concept.	24
3.3 Parity Space Representation.	27
3.4 Continuity Risk Upper Bound in Parity Space.	28
3.5 Continuity Risk due to FANE for Mode k	30
3.6 Integrity Risk Upper Bound in Parity Space.	34
3.7 Fault Mode Plot.	35
4.1 Physical Interpretation of Protection Levels.	42
4.2 Fault Mode Matrix.	44
4.3 Protection Levels and Alert Limits.	54
4.4 MHSS Test Statistics.	57
4.5 Fault Mode Matrix for Exclusion.	60
5.1 H-ARAIM Performance using GPS Only Versus GPS + GLONASS.	64
5.2 H-ARAIM Performance using GPS + GLONASS + Galileo.	65
5.3 V-ARAIM Performance using GPS + GLONASS + Galileo.	66
5.4 Effect of Increasing $P_{const,GPS}$ on H-ARAIM HPLs.	69
5.5 Effect of Degrading GLONASS Performance and Reliability on V-ARAIM Performance.	72

6.1	Number of Satellites Versus Number of Monitored Modes for V-ARAIM.	75
6.2	Satellite Selection Concept.	76
6.3	Performance Degradation when Omitting GPS Satellites Only Versus Omitting the Optimal Number of GPS and/or GLONASS Satellites.	79
6.4	Efficiency Degradation and Performance Improvement when Selecting Satellites Optimally Versus Iteratively.	82
6.5	Percent Change in Efficiency and Performance of V-ARAIM using Satellite Selection Versus Baseline V-ARAIM using GPS + GLONASS + Galileo.	85
6.6	Grouped Fault Mode Matrix.	87
6.7	Percent Improvements in Efficiency and Performance of V-ARAIM using Fault Mode Grouping Versus Baseline V-ARAIM using GPS + GLONASS.	89
D.1	Estimate Error Distribution under Fault-Free Conditions H_0	105
D.2	Estimate Error Distribution under Faulted Conditions H_k	107

ABSTRACT

Since the inception of GPS, satellite navigation has been a widely used means of navigation for both military and civilian users on the ground and in the air. GPS is capable of providing highly accurate positioning and timing information to users around the globe. However, for certain applications, providing high-accuracy position estimates is not sufficient. Because satellites are susceptible to faults, the safety, or integrity, of the position estimates is also of concern, especially in civilian aviation where safety is critical. As such, receiver autonomous integrity monitoring (RAIM) can be used in order to detect and potentially exclude these faults and guarantee the safety of the position estimate. RAIM has been capable of supporting horizontal aircraft navigation using GPS for decades and has proven to be a useful tool. Now, as more global navigation satellite systems (GNSS) become available, the potential for advanced RAIM (ARAIM) to support vertical guidance for aircraft using multiple constellations has become an area of great interest.

In this work, the ARAIM methodology is discussed, and the procedure is outlined, including protection level calculation, fault detection, and exclusion. The procedure is then implemented in a real-time ARAIM prototype. While GPS and Galileo aim to provide worldwide coverage for vertical guidance by 2020 when Galileo is fully operational, ARAIM performance can be examined at present using the current full-strength GPS and GLONASS constellations. This prototype performs position estimation and ARAIM using measurements from the current GPS, GLONASS, and partial Galileo constellations. ARAIM results in a variety of different GNSS scenarios are examined. Furthermore, this work investigates two methods of improving the computational efficiency of the ARAIM algorithm: satellite selection and fault mode grouping.

CHAPTER 1

INTRODUCTION

1.1 GNSS Architecture

The global navigation satellite system (GNSS) architecture is made up of three main components, the control segment, the space segment, and the user segment. The control segment consists of master stations, data upload stations, and monitor stations. Master stations maintain and adjust satellite orbits when necessary. Data upload stations are responsible for transmitting signals to the satellites containing information regarding their clocks, orbits, and the orbits of all other satellites. Monitor stations are responsible for monitoring the health statuses and performance of the satellites in view, and are thus distributed widely over the coverage area of the GNSS.

The space segment consists of all the satellites in the constellation and their payloads, including precise atomic clocks. The satellites are responsible for broadcasting signals to the user segment that can be used for positioning, navigation, and timing (PNT). These signals contain information that identifies the satellite, tells the user where the satellite is (the ephemeris), tells the user its range to the satellite, and gives the user a rough estimate of where other satellites in the constellation are (the almanac).

The user segment consists of both civilian and military users, equipped with GNSS receivers, that receive signals broadcast by the satellites and use them for PNT. The GNSS architecture is shown in Figure 1.1.

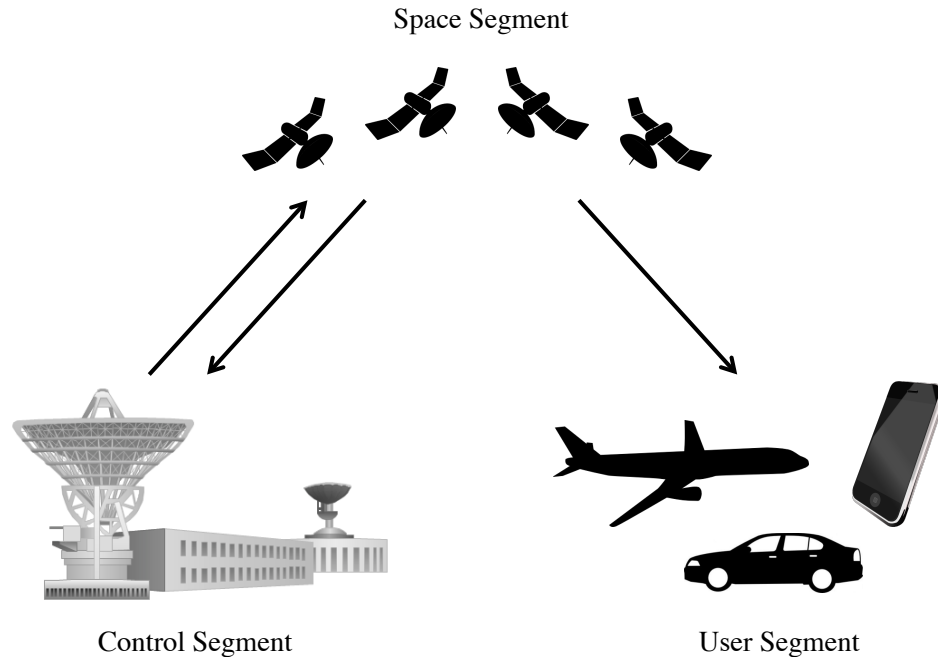


Figure 1.1. GNSS Architecture.

1.2 Current GNSS

1.2.1 GPS. The Global Positioning System (GPS) was conceived by the United States Department of Defense in the early 1970s and is currently maintained by the Air Force Space Command. It is the longest standing and most widely used GNSS, providing global coverage since 1995 [1]. Currently, GPS consists of 24 to 32 satellites in medium Earth orbit (MEO) distributed among six orbital planes at 55 degree inclination. The GPS constellation repeats every sidereal day, meaning that every sidereal day, the same satellites appear in the same locations in the sky to the stationary user.

All GPS satellites broadcast signals on two or more frequencies. On each signal, the carrier modulates a pseudo-random noise (PRN) code identifying the satellite and a navigation message containing orbit and clock parameters. GPS provides two levels of service, the standard positioning service (SPS) and the precise positioning

service (PPS). The SPS is provided by the GPS L1 signal, whose carrier frequency is 1575.42 MHz and lies in the protected aeronautical radio navigation systems (ARNS) frequency band. The GPS L1 coarse acquisition (C/A) code is accessible by civilian users. The PPS is provided by both the L1 and L2 (1227.60 MHz) signals and is accessible only by military users. The precision codes modulated on the L1 and L2 carriers are called the P(Y) codes. Despite the fact that the L2 code is reserved for military use, the L2 carrier is available for civilian use, and is often used in conjunction with the L1 C/A code for dual-frequency applications. While the L2 frequency does not lie in the ARNS band, it can be used as a “stand in” signal for civilians until the GPS L5 signal is fully established. That GPS satellites all broadcast on the same carrier frequencies is made possible by code division multiple access (CDMA) techniques, in which each GPS satellite has a unique PRN code that differentiates it from all other satellites.

1.2.2 GLONASS. Development of the Global Navigation Satellite System (GLONASS) began in the Soviet Union in the late 1970s. Its progress endured the disintegration of the Soviet Union in 1991, as well as severe economic crises in the 1990s, to become a top priority of modern-day Russia. It has been providing global coverage since 2011 and is operated by the Russian Aerospace Defense Forces. It consists nominally of at least 24 MEO satellites distributed among three orbital planes inclined at 65 degrees in order to provide better coverage at higher latitudes. The GLONASS constellation repeats every eight sidereal days.

GLONASS satellites are frequency division multiple access (FDMA), meaning that they all modulate the same standard precision code on different carrier frequencies, along with the navigation message, and thus are differentiated in this manner. The GLONASS L1 carrier frequency for the satellite in slot n is given by

$$f_{L1,n} = f_{L1,0} + n\Delta f_{L1}, \quad (1.1)$$

where $f_{L1,0}$ is the nominal L1 frequency of 1602 MHz, Δf_{L1} is the L1 frequency offset of 562.5 kHz. Similarly, the L2 carrier frequency for satellite n is given by

$$f_{L2,n} = f_{L2,0} + n\Delta f_{L2}, \quad (1.2)$$

where $f_{L2,0}$ is the nominal L2 frequency of 1246 MHz and Δf_{L2} is the L2 frequency offset of 437.5 kHz. The slot number indicates at which position, or slot, the satellite is within the orbital plane. Antipodal satellites (i.e. satellites in opposite slots of the same orbital plane) broadcast on the same carrier frequency since they cannot both be visible to the same user.

Just as for GPS, the GLONASS L1 carrier modulates the C/A and encrypted P(Y) codes, available to civilian and military users, respectively, and the L2 carrier modulates the P(Y) code only. Again, however, civilian users may employ the L2 carrier for dual-frequency applications. The most modern GLONASS satellites, called GLONASS-K, will also transmit CDMA signals in addition to the current FDMA signals.

1.2.3 Galileo. Galileo was created by the European Union (EU) and is operated by the European Space Agency (ESA). Crucially, it is the first GNSS exclusively targeting civilian and commercial use, and it was motivated by the need for a GNSS complementary to but independent from GPS [2]. The constellation repeats every ten sidereal days. Currently, Galileo is not yet at full operational capability (FOC), meaning it is not yet providing global coverage. Once FOC is reached, the constellation will consist of 30 MEO satellites distributed among three orbital planes inclined at 56 degrees.

All Galileo satellites are CDMA, just like GPS satellites, and broadcast on multiple carrier frequencies. Galileo will provide a variety of different services, including the open service (OS) and the commercial service (CS) [2] [3]. The OS is

provided by the freely accessible navigation message (F/NAV) on the E5a carrier and the integrity navigation message (I/NAV) on the E1 and E5b carriers. The I/NAV message contains additional information regarding integrity. The CS is provided by the encrypted commercial navigation message (C/NAV) modulated on the E6 carrier, as well as the I/NAV message. The C/NAV message has not yet been fully specified. Table 1.1 lists the Galileo signals, their frequencies, the navigation messages that they modulate, and the services they provide. The GPS, GLONASS, and Galileo constellations are shown in Figure 1.2.

Table 1.1. Galileo Signals.

Signal	Carrier Frequency (MHz)	Navigation Message	Service
E1	1575.42	I/NAV	OS, CS
E5a	1176.45	F/NAV	OS
E5b	1207.14	I/NAV	OS, CS
E6	1278.75	C/NAV	CS

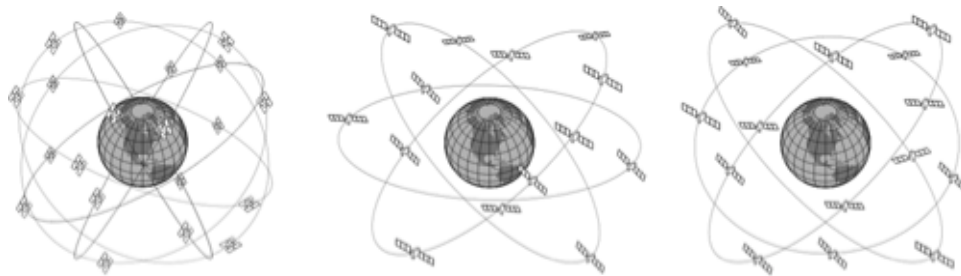


Figure 1.2. GPS (left), Galileo (middle), and GLONASS (right) Constellations.

1.3 RAIM

Satellites are susceptible to faults, which may cause dangerously large positioning errors if left undetected. These faults may come from a variety of sources, both onboard the satellite and from the ground segment. For certain applications, namely aviation, such positioning errors may be intolerable and put the user in a dangerous situation. Receiver autonomous integrity monitoring (RAIM) utilizes redundant ranging measurements in order to detect and potentially exclude faults to ensure the safety of the position estimate generated by the navigation system. Essentially, RAIM detects faults by examining the consistency of the measurements used in the position estimate; inconsistent measurements may be indicative of a fault.

Historically, RAIM has been used to support horizontal guidance for aircraft during en route flight using GPS L1 only [4]. However, the number of GNSS has been increasing with the full deployment of Russia's GLONASS constellation, as well as the strengthening of Europe's Galileo constellation. This has piqued interest in using advanced RAIM (ARAIM), which employs redundant, dual-frequency measurements from multiple GNSS, to support vertical aircraft guidance. While vertical guidance is currently only supported by satellite-based augmentation systems (SBAS), ARAIM aims at achieving worldwide coverage for vertical guidance by 2020 using dual-constellation GPS and Galileo [5]. An important advantage of ARAIM is that, as the name suggests, it is receiver autonomous. As such, ARAIM has the potential to provide coverage to users that are outside SBAS coverage areas or in remote locations where ground-based augmentation systems (GBAS) are not available. Furthermore, the implementation of ARAIM requires almost no additional infrastructure; it relies solely on GNSS measurements, which are processed entirely at the user receiver.

CHAPTER 2

GNSS POSITIONING

Real-time ARAIM begins with a position estimate. Position estimation relies on the concept of trilateration, shown in Figure 2.1, in which the user receives signals from three satellites and computes its location based on the measured range from each satellite. However, because receivers are not usually equipped with precise clocks

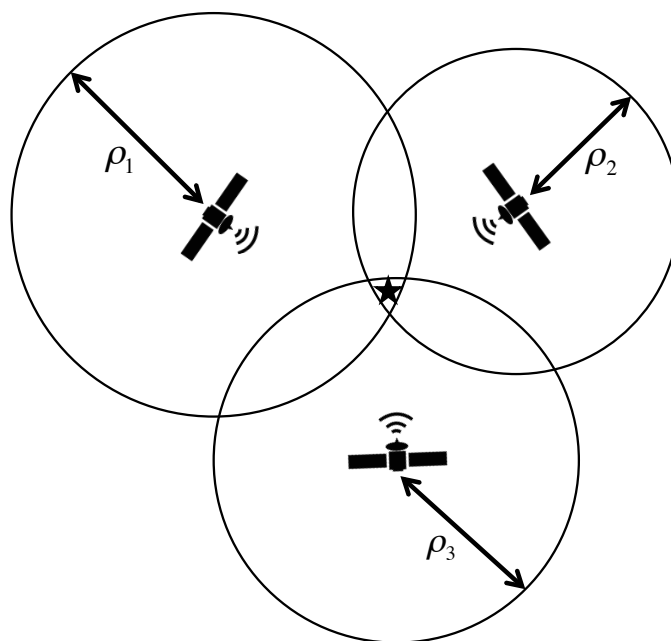


Figure 2.1. Position Estimation by Trilateration.

that are synchronized with the satellite clocks, an additional satellite is required to evaluate this receiver clock offset. When incorporating multiple GNSS into position estimation, one satellite from each constellation must be used to evaluate the receiver clock offset with respect to the system time of that GNSS. Table 2.1 shows the minimum required number of satellites for position estimation.

Table 2.1. Minimum Number of Satellites Required for Estimation.

Number of Constellations	Minimum Number of Satellites
1	4
2	3 + 2
3	2 + 2 + 2

2.1 Code Measurements

The pseudorange is the receiver code measurement made by determining the signal transit time. The pseudorange from the receiver to satellite i is given by

$$\rho_i = r_i + c\delta t_u - c\delta t_{s,i} + cI_{\rho,i} + cT_{\rho,i} + \varepsilon_{\rho,i}, \quad (2.1)$$

where r_i is the true range, δt_u is the receiver clock offset with respect to the GNSS system time, $\delta t_{s,i}$ is the satellite clock offset with respect to GNSS system time, $I_{\rho,i}$ is the ionospheric delay on the code, $T_{\rho,i}$ is the tropospheric delay on the code, $\varepsilon_{\rho,i}$ is the code measurement error, and c is the speed of light in a vacuum. Code measurements are most commonly used in position estimation because they are unambiguous, meaning that the receiver can explicitly determine the range to the satellite by measuring the time elapsed between signal emission from the satellite and signal reception by the receiver.

2.2 Carrier Phase Measurements

The carrier phase is the receiver measurement of the phase of the sinusoidal carrier that modulates the code, and it can also be used for positioning. An advantage of carrier phase measurements is that they are much more precise than code measurements because their signal wavelength is much shorter. For example, the GPS PRN

code that is used to measure the pseudorange has a wavelength of about 300 meters, while the GPS L1 carrier has a wavelength of only about 19 centimeters. However, carrier phase measurements have the distinct disadvantage of being ambiguous; each carrier phase measurement has an associated integer ambiguity, as the receiver is only measuring the instantaneous phase of the incoming signal and cannot know how many complete sinusoidal waves have accumulated since emission from the satellite. The carrier phase measurement model is

$$\lambda_i \phi_i = r_i + c\delta t_u - c\delta t_{s,i} + cI_{\phi,i} + cT_{\phi,i} + \lambda_i N_i + \varepsilon_{\phi,i} \quad (2.2)$$

where $I_{\phi,i}$ is the ionospheric advance on the carrier, λ_i is the carrier wavelength, and N_i is the integer cycle ambiguity. In certain applications, this ambiguity must be resolved. However, the precise nature of the carrier phase measurement can often be exploited when used in conjunction with the code phase measurements, without the need to resolve the cycle ambiguity. The errors associated with the code and carrier phase measurements are discussed next.

2.3 Measurement Errors

A crucial part of the ARAIM algorithm is accurately modeling the measurement errors. These errors primarily consist of signal-in-space (SIS) errors, atmospheric errors, and user errors. Both the accuracy and integrity of the position estimate directly depend on the user's ability to statistically model these errors.

SIS errors are dominated by satellite clock and ephemeris errors. In general, they consist of any errors incurred on board the satellite, including errors due to signal deformation, clock instabilities, and orbit deviations. In reality, the true SIS error is generally not normally distributed, so it is overbounded by a normal distribution. This overbound, shown in Figure 2.2, is called the user ranging accuracy (URA); specifically, the URA is the standard deviation of the SIS error overbound, whose

square can be added to the variances of other errors. All GNSS satellites broadcast a URA (or some variation of it) in the navigation message.

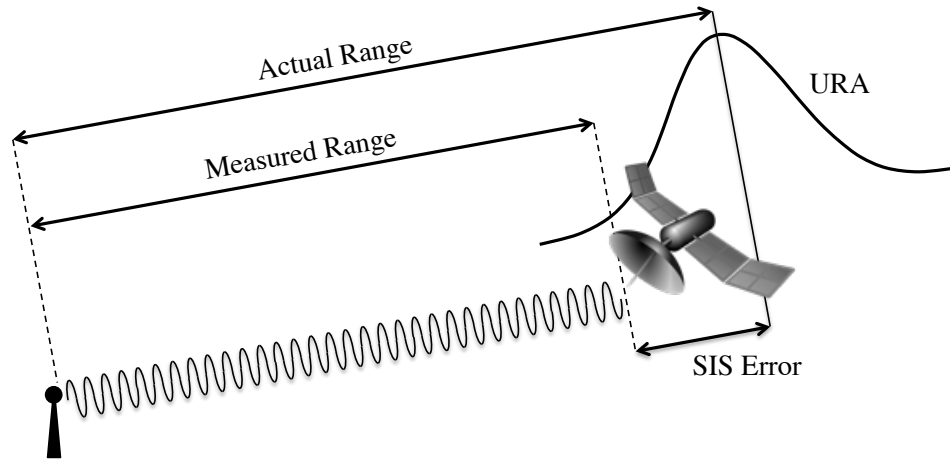


Figure 2.2. Signal-in-Space Error.

Atmospheric errors consist primarily of ionospheric and tropospheric errors. These layers of the atmosphere are shown in Figure 2.3. The ionosphere is the largest

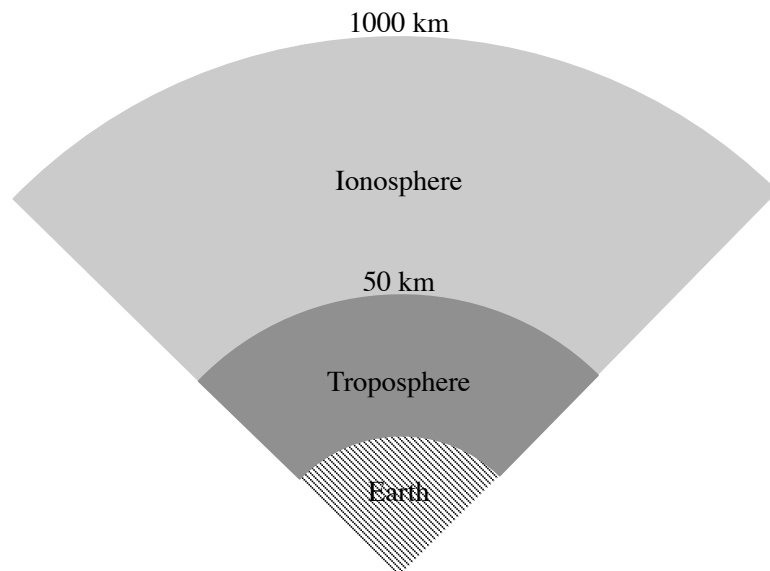


Figure 2.3. Ionosphere and Troposphere.

source of error on GNSS ranging signals, but because ARAIM relies on dual-frequency measurements, the ionospheric error is eliminated explicitly, as will be discussed in the next section. The tropospheric error, however, cannot be eliminated in this manner because the troposphere is a non-dispersive medium; its effects do not depend on signal frequency. Because of this, the tropospheric errors are generally modeled. This work uses a simple model defined in [6] that requires only user location, time of year, and satellite elevation to estimate the tropospheric delay based on average meteorological parameters.

User errors consist of two main sources: receiver noise and multipath. These are user-specific; they depend on the receiver hardware itself and the user's environment. Multipath is a phenomenon in which the same signal reaches the receiver from two or more paths by reflecting off of nearby surfaces, as shown in Figure 2.4. Al-

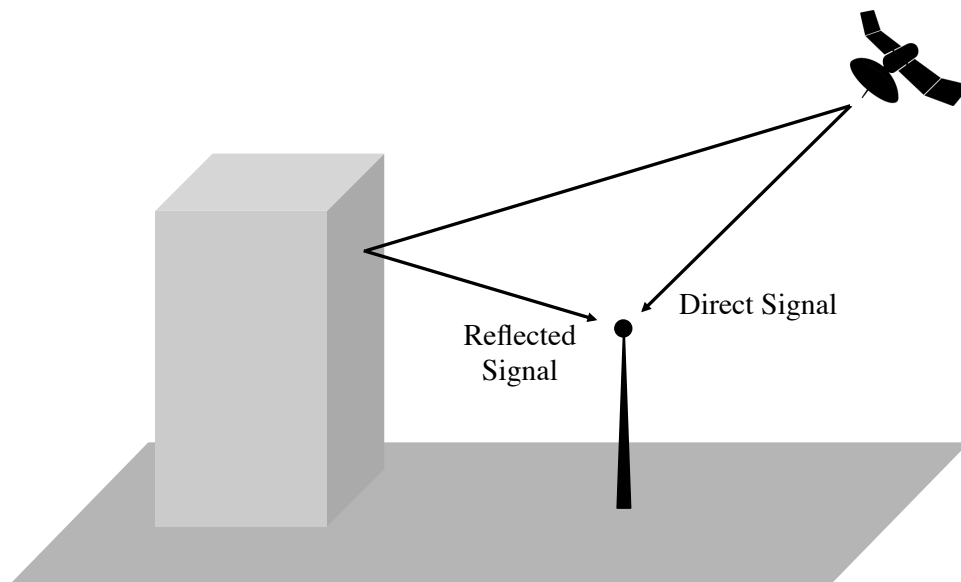


Figure 2.4. Multipath.

though the power of the reflected signal is generally less than the power of the direct signal, the receiver sees the sum of both, resulting in a receiver-perceived ranging

bias. Multipath is generally diminished in an airborne environment, where there are fewer reflective surfaces in the immediate vicinity of the antenna, and its effects can be somewhat mitigated using carrier smoothing, discussed later. However, neither multipath nor receiver noise can be completely eliminated and thus must be modeled accordingly.

Table 2.2 shows the approximate magnitudes of these errors in the raw code and carrier phase measurements, before any steps are taken in their mitigation or elimination.

Table 2.2. GNSS Measurement Error Sources.

Error Source	Approximate Magnitude
Satellite Clock	2 m
Satellite Ephemeris	2 m
Ionosphere	10 m
Troposphere	3 m
Multipath	1 m code / 1 cm carrier
Receiver Noise	0.5 m code / 2 mm carrier

2.4 Ionosphere-Free Measurements

The ionosphere is the largest contributor to overall GNSS ranging errors. Because it is a dispersive medium, it affects the code and the carrier in different ways; specifically, it advances the carrier and delays the code by equal amounts [1]. This amount is simply a function of the carrier frequency and the total electron content (TEC), which quantifies the amount of electrons along the signal path. As such, the

carrier advance and code delay are different for signals of different frequencies (the GPS L1 and L2 signals, for example). Because of this, multiple signals broadcast from the same satellite on different carrier frequencies can be utilized to eliminate the first-order ionospheric effects explicitly. This is done by forming a linear combination of the two measurements from each satellite. The ionosphere-free code measurement is given by

$$\rho_{IF} = \frac{f_{L1}^2}{f_{L1}^2 - f_{L2}^2} \rho_{L1} - \frac{f_{L2}^2}{f_{L1}^2 - f_{L2}^2} \rho_{L2}, \quad (2.3)$$

and the ionosphere-free carrier phase measurement (in cycles) is given by

$$\phi_{IF} = \frac{f_{L1}^2}{f_{L1}^2 - f_{L2}^2} \phi_{L1} - \frac{f_{L2}^2}{f_{L1}^2 - f_{L2}^2} \phi_{L2}. \quad (2.4)$$

The carrier measurement can be more conveniently expressed in units of distance, as

$$\Phi_{IF} = \frac{f_{L1}^2}{f_{L1}^2 - f_{L2}^2} \lambda_{L1} \phi_{L1} - \frac{f_{L2}^2}{f_{L1}^2 - f_{L2}^2} \lambda_{L2} \phi_{L2}, \quad (2.5)$$

where λ_{L1} and λ_{L2} are the wavelengths of the L1 and L2 carriers, respectively, or any other two signals used to form the ionosphere-free combination for that matter. By using ionosphere-free measurements, the terms $cI_{\rho,i}$ and $cI_{\phi,i}$ may be eliminated from equations 2.1 and 2.2, respectively. While the ionospheric effects are eliminated, receiver noise is amplified in the ionosphere-free measurements [1]. Despite this, using ionosphere-free measurements is almost always worth this amplification in noise.

2.5 Carrier Smoothing

As mentioned previously, the code is often used in positioning because it is unambiguous. However, the code is highly susceptible to multipath because its chip length is so long - about 300 meters for GPS. Thus, a low-pass filter, shown in Figure 2.5, may be implemented to suppress the high-frequency code noise by incorporating carrier phase measurements, which are relatively smooth.

The carrier smoothed code utilizes current and past measurements in the form

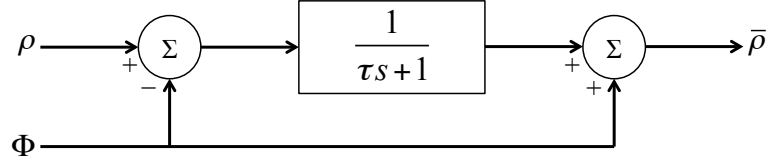


Figure 2.5. Carrier Smoothing Block Diagram.

of a recursive relationship, given in discrete form by

$$\bar{\rho}_t = \frac{1}{M}\rho_t + \frac{M-1}{M}[\bar{\rho}_{t-1} + (\Phi_t - \Phi_{t-1})], \quad (2.6)$$

where

$$M = \frac{\tau}{\delta t}. \quad (2.7)$$

Equation 2.6 is derived in Appendix A. Here, $\bar{\rho}$ is the smoothed code, t is the time epoch, τ is the smoothing time constant, and δt is the sample time interval. In general, the time constant is limited to about 100 seconds due to ionospheric divergence [6], but when using ionosphere-free measurements, it is possible to smooth indefinitely (i.e. $\tau = \infty$). As the time constant becomes larger, more weight is placed on the smoothed code from the previous epoch and the carrier phase difference, and less on the current raw code measurement. Note that no cycle ambiguity resolution is necessary because it is only the time difference in carrier measurements that is being used. As long as the receiver maintains lock on the carrier, the integer ambiguity, although unknown, does not change (unless a cycle slip occurs, which is not discussed here). If, however, the receiver loses lock on a satellite, smoothing must be reinitialized because the integer ambiguity will likely be different once lock is reacquired.

2.6 Satellite Orbit Computation

GPS and Galileo satellite clock corrections are both for the dual-frequency user. In other words, the clock correction parameters broadcast in the navigation message are in reference to ionosphere-free time, which is the (fictitious) time of signal

emission of the ionosphere-free measurements. This means that the dual-frequency user can apply the satellite clock corrections

$$\delta t_s = a_{f0} + a_{f1} (t_{se} - t_{oc}) + a_{f2} (t_{se} - t_{oc})^2 - \frac{2e\sqrt{\mu A} \sin E}{c^2} \quad (2.8)$$

defined in [7] and [3] directly to the ionosphere-free measurements themselves. Here, t_{se} is the time of signal emission of the ionosphere-free signal and the last term in the equation is the relativistic correction term. The orbital eccentricity e , semi-major axis A , and the clock parameters a_{f0} , a_{f1} , a_{f2} , and t_{oc} are all contained in the navigation message. Table 2.3 lists the ephemeris and clock parameters contained in the GPS and Galileo navigation messages. Because the clock correction parameters contained in the GPS navigation message are for L1/L2 users, L1/L5 users must take additional steps to correct for the satellite clock bias when using L1/L5 service. Galileo provides multiple dual-frequency services, including E1/E5a and E1/E5b provided by the F/NAV and I/NAV messages, respectively. As such, the clock correction parameters contained in these navigation messages describe the ionosphere-free E1/E5a and ionosphere-free E1/E5b signals, respectively.

Orbit computation for GPS and Galileo is performed following the procedures outlined in the IS-GPS-200 [7] and the Galileo Interface Control Document (ICD) [3], respectively. GPS satellite locations are computed in the WGS-84 frame, which is an Earth-centered, Earth-fixed (ECEF) frame, and Galileo satellite locations are computed in the Galileo Terrestrial Reference Frame (GTRF), which is also an ECEF frame and is coincident with WGS-84. As such, no transformation needs to be made between GTRF and WGS-84 in order to perform position estimation using GPS and Galileo.

GLONASS provides different parameters in the navigation message than GPS or Galileo both for orbit computation and satellite clock correction, as shown in Table 2.4. The GLONASS navigation message contains clock correction parameters

Table 2.3. GPS and Galileo Clock and Ephemeris Parameters.

Field	Description
t_{oe}	Reference time of ephemeris
M_0	Mean anomaly at t_{oe}
Δn	Mean motion difference
e	Orbital eccentricity
\sqrt{A}	Square root of semi-major axis
Ω_0	Longitude of ascending node
i_0	Inclination angle at t_{oe}
ω	Argument of perigee
$\dot{\Omega}$	Rate of change of right ascension
\dot{i}	Rate of change of inclination angle
$C_{uc}, C_{us}, C_{rc}, C_{rs}, C_{ic}, C_{is}$	Harmonic corrections
t_{oc}	Clock correction data reference time
a_{f0}, a_{f1}, a_{f2}	Satellite clock correction coefficients

for both the L1 and the L2 measurements, referenced to the times of L1 and L2 signal emission, respectively. The crucial implication of this is that the satellite clock biases must be accounted for in the raw L1 and L2 measurements before combining them to form an ionosphere-free combination. The satellite clock biases for the L1 and L2 measurements are

$$\delta t_{s,L1} = -\tau + \gamma (t_{se,L1} - t_{oe}) \quad (2.9)$$

Table 2.4. GLONASS Clock and Ephemeris Parameters.

Field	Description
t_{oe}	Reference time of ephemeris
x_0, y_0, z_0	Initial satellite locations in PZ-90 at t_{oe}
$\dot{x}_0, \dot{y}_0, \dot{z}_0$	Initial satellite velocities in PZ-90 at t_{oe}
$\ddot{x}_{LS}, \ddot{y}_{LS}, \ddot{z}_{LS}$	Initial lunisolar accelerations in PZ-90 at t_{oe}
τ	Satellite clock offset with respect to GLONASS system time at t_{oe}
$\Delta\tau$	Difference between L1 and L2 signal emission times
γ	Relative deviation of carrier frequency at t_{oe}

and

$$\delta t_{s,L2} = -\tau + \gamma (t_{se,L1} - t_{oe}) + \Delta\tau, \quad (2.10)$$

respectively [8], where $t_{se,L1}$ is the time of signal emission of the L1 signal, τ is the satellite clock offset with respect to GLONASS system time at the reference time of ephemeris t_{oe} , γ is the relative deviation of the carrier frequency from the nominal frequency at the time of ephemeris, and $\Delta\tau$ is the time difference between the L1 and L2 signals. The carrier frequency deviation is given by

$$\gamma = \frac{f - f_{nom}}{f_{nom}} \quad (2.11)$$

in [9], where f is the predicted carrier frequency at the time of ephemeris and f_{nom} is the nominal carrier frequency, given in equations 1.1 and 1.2. The time difference between the L1 and L2 signals is given by

$$\Delta\tau = t_{L2} - t_{L1} \quad (2.12)$$

in [9], where t_{L1} and t_{L2} are the hardware delays on the L1 and L2 signals, respectively, sustained aboard the satellite.

GLONASS satellite positions are computed by propagating a set of initial positions, velocities, and accelerations from the reference time of ephemeris to the desired epoch within the ephemeris interval via numerical integration. This procedure is outlined in Appendix B following the GLONASS ICD [9]. GLONASS satellite locations are calculated in the PZ-90 frame, which is an ECEF frame but is not coincident with WGS-84. Thus, the GLONASS satellite coordinates in PZ-90 must be transformed into coordinates in the WGS-84 frame in order to use both GPS and GLONASS measurements in position estimation. The transformations between various realizations of PZ-90 and WGS-84 can be found in [10].

2.7 Position Estimation

For GPS and Galileo measurements, the ionosphere-free pseudoranges are formed from the L1 and L2 pseudoranges according to equation 2.3. Then, for each satellite, the satellite clock bias, given in equation 2.8, is computed at the time of ionosphere-free signal emission, which is given by

$$t_{se} = t - \frac{\rho_{IF}}{c}, \quad (2.13)$$

where t is the epoch at which a position estimate is desired and ρ_{IF} is the carrier-smoothed, ionosphere-free pseudorange with the tropospheric corrections applied. The satellite clock bias is accounted for in the pseudorange as shown in equation 2.1. Now, the tropospheric delay, ionospheric delay, and satellite clock bias have been accounted for in the pseudorange, leaving only the receiver clock bias remaining.

For GLONASS, the satellite clock biases are computed at the times of L1 and L2 signal emission (calculated via equation 2.13 by replacing ρ_{IF} with ρ_{L1} or ρ_{L2} , as appropriate) according to equations 2.9 and 2.10. The satellite clock biases

are applied to the pseudoranges as shown in equation 2.1, and then the ionosphere-free combination is formed according to equation 2.3. Again, the tropospheric delay, ionospheric delay, and satellite clock bias have been eliminated from the pseudorange equation.

Now, the pseudorange measurement equation is

$$\rho_{IF,i} = r_i + c\delta t_u + \nu_i \quad (2.14)$$

for all satellites i . The error term ν_i consists of normally distributed clock and ephemeris errors, residual tropospheric error, and multipath and receiver noise.

Next, the satellite locations are computed at the times of ionosphere-free signal emission as discussed previously. These locations are most conveniently expressed in an Earth-centered, Earth-fixed (ECEF) frame, and it is customary to use the WGS-84 frame. Because the Earth rotates about its axis during the signals' transit times, the Earth must be rotated to its orientation at the times of signal emission. For each satellite, the Earth rotates by

$$\Omega_i = \omega_E \frac{\rho_{IF,i} - c\delta t_u}{c} \quad (2.15)$$

radians during the signal transit time, where ω_E is the rotation rate of Earth, in radians per second, and δt_u is the receiver time offset with respect to the the GNSS system time of satellite i . Thus, at the time of signal emission, the location of satellite i is

$$\xi_i = \begin{bmatrix} \cos \Omega_i & \sin \Omega_i & 0 \\ -\sin \Omega_i & \cos \Omega_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \tilde{\xi}_i \quad (2.16)$$

in the WGS-84 frame, where $\tilde{\xi}_i$ is a column vector of the x , y , and z components of the location of satellite i in the WGS-84 frame before the Earth rotation has been taken into account.

Now that the satellite locations and pseudoranges have been computed for n GPS satellites, m GLONASS satellites, and l Galileo satellites, the measurement model is constructed as

$$z = Gx + \nu, \quad (2.17)$$

where

$$z = \begin{bmatrix} \rho_{GPS,1} - e_{GPS,1}^T \xi_{GPS,1} \\ \vdots \\ \rho_{GPS,n} - e_{GPS,n}^T \xi_{GPS,n} \\ \rho_{GLO,1} - e_{GLO,1}^T \xi_{GLO,1} \\ \vdots \\ \rho_{GLO,m} - e_{GLO,m}^T \xi_{GLO,m} \\ \rho_{GAL,1} - e_{GAL,1}^T \xi_{GAL,1} \\ \vdots \\ \rho_{GAL,l} - e_{GAL,l}^T \xi_{GAL,l} \end{bmatrix} \quad (2.18)$$

is the measurement vector,

$$G = \begin{bmatrix} -e_{GPS,1}^T & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ -e_{GPS,n}^T & 1 & 0 & 0 \\ -e_{GLO,1}^T & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ -e_{GLO,m}^T & 0 & 1 & 0 \\ -e_{GAL,1}^T & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -e_{GAL,l}^T & 0 & 0 & 1 \end{bmatrix} \quad (2.19)$$

is the observation/geometry matrix, and

$$x = \begin{bmatrix} x \\ y \\ z \\ \delta t_{u,GPS} \\ \delta t_{u,GLO} \\ \delta t_{u,GAL} \end{bmatrix} \quad (2.20)$$

is the solution, consisting of three position coordinates x , y , and z , and a receiver clock offset for each GNSS. The noise term

$$\nu \sim N(0, C) \quad (2.21)$$

represents the zero mean, normally distributed random errors due to clock and

ephemeris errors, tropospheric error, multipath, and receiver noise with covariance C , which will be discussed quantitatively later.

In practice, the solution \hat{x} can be found iteratively using least-squares estimation. Generally, a good initial guess is not necessary, but it is straightforward to use the final estimate from the previous time epoch as the initial guess for the current estimate. The unit line-of-sight vectors to each satellite are computed as

$$e_i = \frac{\xi_i - \hat{x}_{prev}}{\|\xi_i - \hat{x}_{prev}\|}, \quad (2.22)$$

where \hat{x}_{prev} is a column vector of the position components of the estimate from the previous iteration. The least-squares solution is

$$\hat{x} = (G^T W G)^{-1} G^T W z, \quad (2.23)$$

where $W = C^{-1}$ weights each measurement by its variance. The pseudo-inverse is defined as

$$S = (G^T W G)^{-1} G^T W, \quad (2.24)$$

and will be used later. The solution \hat{x} is used as the initial guess \hat{x}_{prev} in the next iteration at the current epoch, and this iteration is continued until the difference between the position components of \hat{x} and \hat{x}_{prev} is sufficiently small. The difference between the receiver clock offsets at each iteration can be ignored because these are only nuisance parameters, and because the measurement model is linear in δt_u . Because the satellite locations have been expressed in the WGS-84 frame, the position solution is also expressed in WGS-84 coordinates. The user may then convert these coordinates into the more intuitive latitude, longitude, altitude (LLA) frame.

CHAPTER 3
METHODOLOGY

3.1 Multiple Hypothesis Solution Separation

The real-time ARAIM procedure begins with a position estimate, computes protection levels, which quantify the integrity of the position solution, and performs fault detection and exclusion (FDE) at each GNSS measurement epoch. This procedure is summarized by the flow chart in Figure 3.1. This work employs a particu-

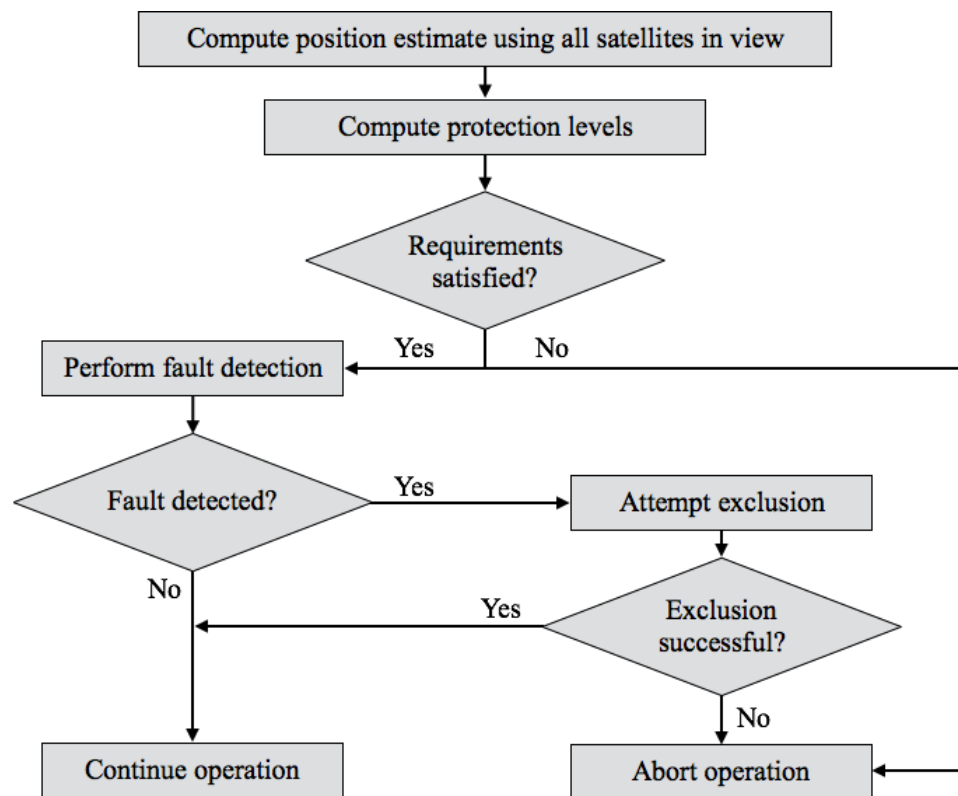


Figure 3.1. Real-Time ARAIM Implementation Flow Chart.

lar type of ARAIM called multiple hypothesis solution separation (MHSS) ARAIM. Essentially, MHSS ARAIM hypothesizes faults on different subsets of satellites, or

modes. For each mode, test statistics are compared to corresponding detection thresholds in order to detect whether or not the faults that were hypothesized are actually present in the system. The way these test statistics and detection thresholds are computed will be discussed later. In Figure 3.2, the fault being monitored is a fault on the second satellite, as indicated by the white signal. This is an example of one of the many fault modes that the MHSS ARAIM algorithm monitors.

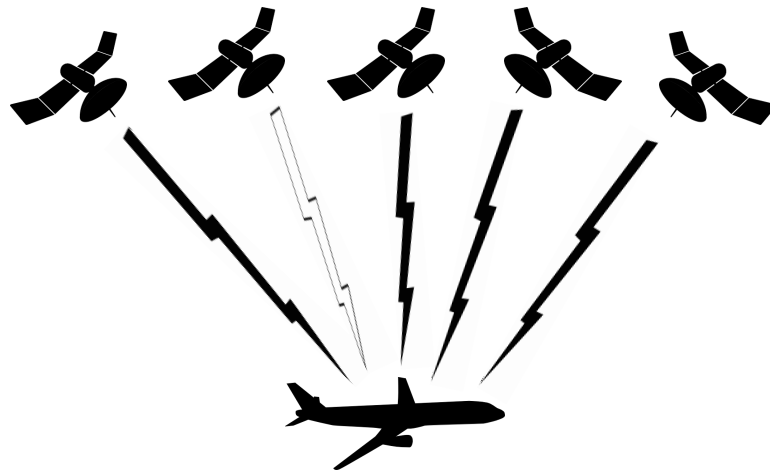


Figure 3.2. MHSS Concept.

3.2 Continuity Risk

Continuity is the ability of a system to perform its intended operation without unscheduled interruption. These interruptions happen when a false alarm occurs and cannot be excluded and when a fault is detected and cannot be excluded. While the detection function cannot differentiate between false alarm and true detection, it is useful in recovering continuity after a fault has been detected, whether it be a true fault or a false alarm. However, frequent detections without the exclusion function pose a continuity threat, and frequent detections with the exclusion function mean that it is constantly called upon, which degrades integrity due to the possibility of excluding the wrong satellite(s). Therefore, the continuity risk is limited by the

continuity risk requirement, as

$$C_{req} = P_{FANE} + P_{FDNE}, \quad (3.1)$$

where P_{FANE} is the probability of false alarm, no exclusion (FANE) and P_{FDNE} is the probability of fault detection, no exculsion (FDNE) [11]. The FANE term is given by

$$P_{FANE} = P \left(\bigcup_{k=1}^n |q_k| > T_k \cap \bigcap_{e=1}^{n_e} \left(\bigcup_{\substack{j=1 \\ j \neq e}}^n |q_{e,j}| > T_{e,j} \right) \middle| H_0 \right) P_{H_0}, \quad (3.2)$$

and the FDNE term is given by

$$P_{FDNE} = \sum_{k=1}^n P \left(\bigcup_{k=1}^n |q_k| > T_k \cap \bigcap_{e=1}^{n_e} \left(\bigcup_{\substack{j=1 \\ j \neq e}}^n |q_{e,j}| > T_{e,j} \right) \middle| H_k \right) P_{H_k}. \quad (3.3)$$

In the above equations, k indexes the monitored fault mode, e indexes the mode that is attempting to be excluded, j indexes the second layer monitored mode (i.e. those remaining modes that must be monitored after omitting a mode in an attempt to exclude it), q_k and T_k are the first layer detection test statistics and detection thresholds for mode k , respectively, $q_{e,j}$ and $T_{e,j}$ are the second layer detection test statistics and exclusion thresholds for mode j after attempting to exclude mode e , respectively, and H_0 and H_k indicate the fault-free mode and fault mode k , respectively. The symbol $\bigcup_{k=1}^n$ reads as “for any k from 1 to n ,” the symbol $\bigcap_{e=1}^{n_e}$ reads as “for all e from 1 to n_e ,” and \cap indicates the intersection. Thus, the FANE and FDNE terms are the probability that a fault is detected (i.e. $|q_k| > T_k$) and cannot be excluded since, for all exclusion options e , at least one second layer test statistic exceeds its exclusion threshold (i.e. $\bigcap_{e=1}^{n_e} \left(\bigcup_{\substack{j=1 \\ j \neq e}}^n |q_{e,j}| > T_{e,j} \right)$), under fault-free conditions H_0 and faulted conditions H_k , respectively.

First, the FANE term can be bounded by ignoring knowledge of any second

layer detection, as

$$P \left(\bigcup_{k=1}^n |q_k| > T_k \cap \bigcap_{e=1}^{n_e} \left(\bigcup_{\substack{j=1 \\ j \neq e}}^n |q_{e,j}| > T_{e,j} \right) \middle| H_0 \right) P_{H_0} \leq P \left(\bigcup_{k=1}^n |q_k| > T_k \middle| H_0 \right) P_{H_0}. \quad (3.4)$$

This is a tight upper bound because the probability of second layer detection after attempting to exclude a false alarm is large. Now, the continuity risk due to FANE is simply

$$P_{FANE} \leq P(|q_1| > T_1 \cup |q_2| > T_2 \cup \dots \cup |q_n| > T_n | H_0) P_{H_0}, \quad (3.5)$$

which is the probability of any test statistic exceeding its threshold under fault-free conditions. Here, \cup indicates the union. This false alarm bound can be visualized using a parity space representation, which is useful because the various fault modes can be represented as lines in the space, and a fault as a vector in the space. Here, a simple example will be presented according to [12] in which a single state x is to be estimated using three measurements according to the linear model

$$z = Hx + f + \nu, \quad (3.6)$$

where z is the 3×1 measurement vector, H is the observation matrix given by

$$H = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T, \quad (3.7)$$

f is the 3×1 fault vector, and $\nu \sim N(0, I)$ is the measurement noise. The parity space, or left null space, of the matrix H consists of all vectors y such that $y^T H = 0$, and is of dimension $n - m$, where n is the number of measurements and m is the number of states to be estimated. For this example, parity space is two-dimensional, which is convenient for graphical representation. Figure 3.3 shows the three fault modes corresponding to each measurement, as well as an arbitrary fault vector projected into parity space, called the parity vector, denoted as p . It has been shown in

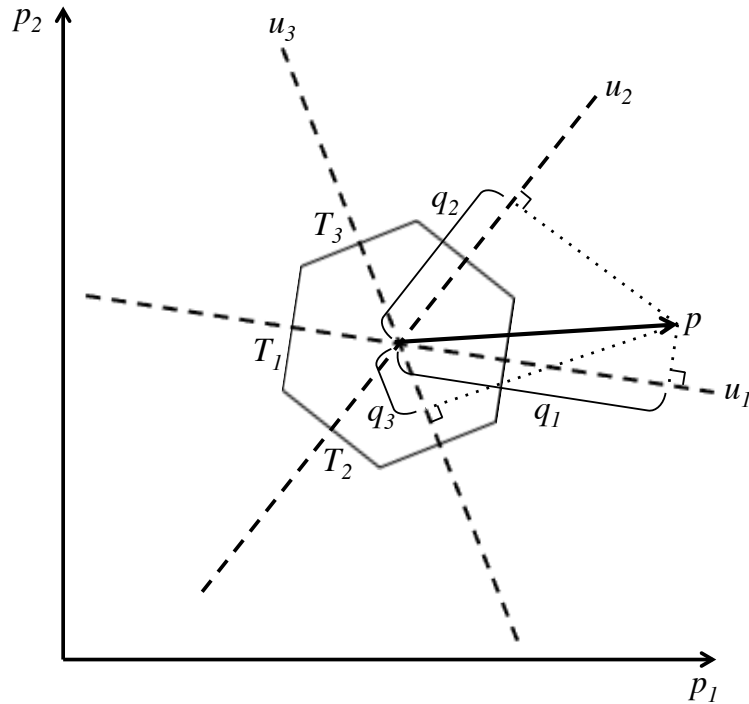


Figure 3.3. Parity Space Representation.

[12] that the solution separation test statistics q_k are simply the projections of the parity vector onto the fault mode lines u_k . If the parity vector lies inside the hexagon, no fault is detected because no test statistic exceeds its threshold. Conversely, the detection region is the area outside the hexagon, which corresponds to the continuity risk due to FANE in equation 3.5 under fault-free conditions. Not shown in Figure 3.3 are the lines of constant probability density, which, under fault-free conditions H_0 , are circles centered around the origin of the parity vector. The probability of the parity vector lying outside the hexagon under H_0 is equal to the continuity risk requirement.

However, the expression in equation 3.5 is very difficult to evaluate in practice because it would require integration over an n -dimensional joint probability density function since the test statistics are correlated. Therefore, the continuity risk due to

FANE is further upper-bounded by

$$P_{FANE} \leq \sum_{k=1}^n P(|q_k| > T_k | H_0) P_{H_0}. \quad (3.8)$$

Essentially, this upper bound indicates that the continuity risk due to FANE for each mode is only due to the test statistic exceeding the detection threshold for that mode. This formulation allows the total FANE contribution to be allocated among the modes, as the upper bound implies the mutual independence of the test statistics, and thus it can be determined analytically. Because the test statistics are not mutually independent in reality, this is a conservative upper bound. This conservatism is demonstrated in Figure 3.4 using a parity space representation for the linear model in equation 3.6. Again, the no detection region is the white hexagon.

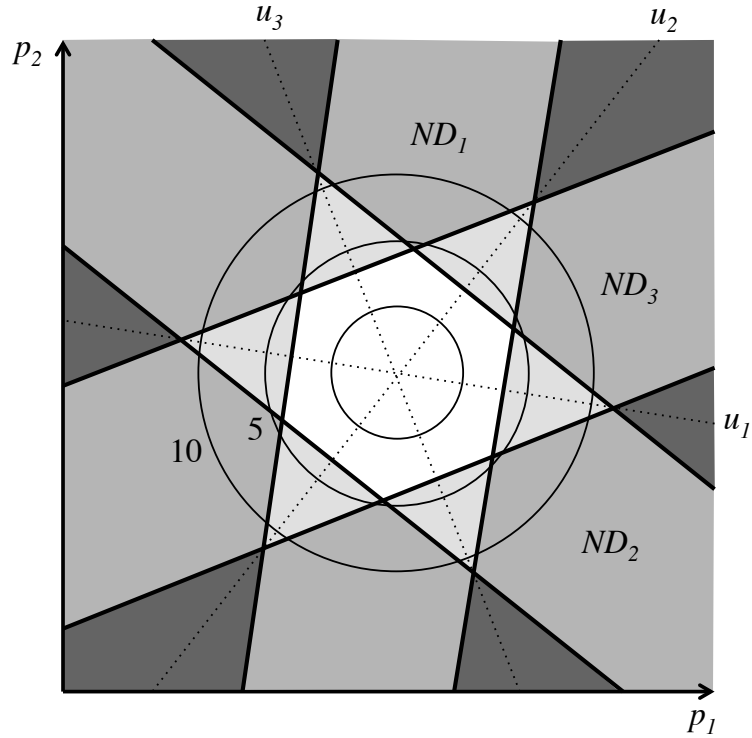


Figure 3.4. Continuity Risk Upper Bound in Parity Space.

The stripes labelled ND_k are perpendicular to their respective fault mode lines u_k

and indicate regions in which the test statistic q_k is less than the threshold T_k for that particular mode. If the parity vector lies in any of the light gray triangles on the edge of the no detection region, the continuity risk is counted once since only one test statistic exceeds its threshold. However, if the parity vector lies in the medium gray areas, the continuity risk is counted twice because two test statistics exceed their thresholds. Furthermore, if the parity vector lies in the dark gray areas, the continuity risk is counted three times because all three test statistics exceed their thresholds. In general, if there are n measurements, there may be regions in which the continuity risk due to FANE is counted n times. This over-counting is why equation 3.8 represents a conservative upper bound. It is important to note that this bound may still be considered a tight bound because the probability of the parity vector lying in any of the over-counted regions under fault-free conditions is small, as indicated by the circles of constant probability density in Figure 3.4. The contours are labelled as $-\log_{10} \text{pdf}(p)$, where $\text{pdf}(p)$ is the probability density function of the (bivariate) parity vector [12]. However, as the number of monitored modes increases, this bound becomes looser; this will be discussed later.

The conservative nature of equation 3.8 may also be represented in a more intuitive manner using a simple example. Consider events A and B. The probability of either A or B occurring is given by

$$P(A \cup B) = P(A) + P(B) - P(A \cap B). \quad (3.9)$$

If A and B are independent, then $P(A \cap B) = 0$. However, if they are not independent, then $P(A \cap B) \neq 0$. The right side of equation 3.5 is analogous to $P(A \cup B)$, where A and B signify events in which one of the test statistics exceeds its threshold. The right side of equation 3.8, which is the upper bound on the continuity risk due to FANE, corresponds to $P(A) + P(B)$. It is clear that

$$P(A) + P(B) \geq P(A \cup B), \quad (3.10)$$

and therefore

$$\sum_{k=1}^n P(|q_k| > T_k | H_0) \geq P(|q_1| > T_1 \cup |q_2| > T_2 \cup \dots \cup |q_n| > T_n | H_0). \quad (3.11)$$

With the bound shown in equation 3.8 applied, the detection thresholds T_k can be set individually to limit the continuity risk due to FANE from each of the monitored fault modes, as shown in Figure 3.5. The test statistic is plotted on the x -axis, and

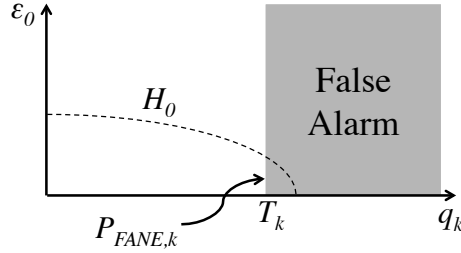


Figure 3.5. Continuity Risk due to FANE for Mode k .

the estimation error is plotted on the y -axis. It has been shown in [12] that the estimation error and the test statistic are independent.

The FDNE contribution from equation 3.3 is bounded next. First, knowledge of first layer detection is omitted, as

$$\begin{aligned} \sum_{k=1}^n P \left(\bigcup_{k=1}^n |q_k| > T_k \cap \bigcap_{e=1}^{n_e} \left(\bigcup_{\substack{j=1 \\ j \neq e}}^n |q_{e,j}| > T_{e,j} \right) \middle| H_k \right) P_{H_k} \\ \leq \sum_{k=1}^n P \left(\bigcap_{e=1}^{n_e} \left(\bigcup_{\substack{j=1 \\ j \neq e}}^n |q_{e,j}| > T_{e,j} \right) \middle| H_k \right) P_{H_k}. \end{aligned} \quad (3.12)$$

This may be considered a tight bound because the probability of detecting a fault when one is present is nearly one. The right side of this equation is then the sum over all modes of the probability that any second layer test statistic exceeds its exclusion threshold for every exclusion attempt. This may be further upper bounded by only

considering the case in which the faulted mode is the one attempting to be excluded (correct exclusion, i.e. $e = k$), as

$$\sum_{k=1}^n P \left(\bigcap_{e=1}^{n_e} \left(\bigcup_{\substack{j=1 \\ j \neq e}}^n |q_{e,j}| > T_{e,j} \right) \middle| H_k \right) P_{H_k} \leq \sum_{k=1}^n P \left(\bigcup_{\substack{j=1 \\ j \neq k}}^n |q_{k,j}| > T_{k,j} \middle| H_0 \right) P_{H_k}. \quad (3.13)$$

First, note that now the second layer test statistics are fault-free, since the faulted mode is omitted, and are thus described by the distribution H_0 . However, this probability is still conditioned on the fact that the fault occurred in the first place, and thus is multiplied by P_{H_k} . This bound is indeed conservative because knowledge of the second layer detection tests for the wrong exclusion cases has been ignored. It is also a tight bound because, if the wrong mode is omitted in an attempt to exclude it, there is a high probability that any of the second layer test statistics exceed its exclusion threshold, as the fault is still present. The right side of equation 3.13 can also be written as

$$P_{FDNE} \leq \sum_{k=1}^n P (|q_{k,1}| > T_{k,1} \cup |q_{k,2}| > T_{k,2} \cup \dots \cup |q_{k,n}| > T_{k,n} | H_0) P_{H_k}. \quad (3.14)$$

This can be upper bounded in exactly the same way as equation 3.5 was (recall equation 3.9):

$$\sum_{k=1}^n P (|q_{k,1}| > T_{k,1} \cup |q_{k,2}| > T_{k,2} \cup \dots \cup |q_{k,n}| > T_{k,n} | H_0) P_{H_k} \leq \sum_{k=1}^n \sum_{\substack{j=1 \\ j \neq k}}^n P (|q_{k,j}| > T_{k,j} | H_0) P_{H_k}. \quad (3.15)$$

Again, this bound implies the mutual independence of the second layer modes (even though they are not independent in reality), and thus enables the FDNE contribution to be distributed among them. In summary, the continuity risk due to FANE and FDNE is upper bounded as

$$C_{req} \leq \sum_{k=1}^n P (|q_k| > T_k | H_0) P_{H_0} + \sum_{k=1}^n \sum_{\substack{j=1 \\ j \neq k}}^n P (|q_{k,j}| > T_{k,j} | H_0) P_{H_k}. \quad (3.16)$$

The first term on the right side of this equation will be used to set the detection thresholds, and the second term will be used to set the exclusion thresholds. This will be discussed later.

3.3 Integrity Risk

Integrity risk is defined as the probability of an undetected system error or fault causing hazardously misleading information (HMI), which may manifest itself as a large positioning error. Integrity risk arises from many sources, including HMI under fault-free conditions, HMI due to missed fault detection, HMI due to fault detection and correct exclusion, and HMI due to fault detection and wrong exclusion. The integrity risk of a system that performs both detection and exclusion is

$$P_{HMI} = \sum_{k=0}^n P \left(|\varepsilon_0| > L \cap \left(\bigcap_{k=1}^n |q_k| < T_k \right) \middle| H_k \right) P_{H_k} + \sum_{e=1}^{n_e} \sum_{j=0}^n P \left(|\varepsilon_e| > L \cap \left(\bigcap_{\substack{j=1 \\ j \neq e}}^n |q_{e,j}| < T_{e,j} \right) \cap \left(\bigcup_{k=1}^n |q_k| > T_k \right) \middle| H_j \right) P_{H_j} \quad (3.17)$$

per [11]. In this equation, L denotes an as yet undefined limit on the estimation error; it will be discussed later. Note that the first term, which is the integrity risk from the detection function, includes the integrity risk from the fault-free mode $k = 0$. Likewise, the second term, which is the integrity risk from the exclusion function, includes the integrity risk associated with correct exclusion. Just as with the continuity risk, both terms in the integrity risk equation will be addressed individually.

First, the integrity risk due to the detection function is upper bounded. It can

be split into two terms, the fault-free term and the sum of the faulted terms, as

$$\begin{aligned} \sum_{k=0}^n P \left(|\varepsilon_0| > L \cap \left(\bigcap_{k=1}^n |q_k| < T_k \right) \middle| H_k \right) P_{H_k} = \\ P \left(|\varepsilon_0| > L \cap \left(\bigcap_{k=1}^n |q_k| < T_k \right) \middle| H_0 \right) P_{H_0} + \\ \sum_{k=1}^n P \left(|\varepsilon_0| > L \cap \left(\bigcap_{k=1}^n |q_k| < T_k \right) \middle| H_k \right) P_{H_k}. \end{aligned} \quad (3.18)$$

The fault-free term can then be bounded by ignoring knowledge of the test statistics, as

$$P \left(|\varepsilon_0| > L \cap \left(\bigcap_{k=1}^n |q_k| < T_k \right) \middle| H_0 \right) P_{H_0} \leq P(|\varepsilon_0| > L | H_0) P_{H_0}. \quad (3.19)$$

This is a tight bound because, as shown in Figure 3.4, the probability of the parity vector lying in the white hexagon, meaning that no fault is detected, under H_0 is large. Next, the faulted terms can be upper bounded as

$$\begin{aligned} \sum_{k=1}^n P \left(|\varepsilon_0| > L \cap \left(\bigcap_{k=1}^n |q_k| < T_k \right) \middle| H_k \right) P_{H_k} \leq \\ \sum_{k=1}^n P(|\varepsilon_0| > L \cap |q_k| < T_k | H_k) P_{H_k}. \end{aligned} \quad (3.20)$$

Essentially, this ignores knowledge of all other test statistics except for the one specifically designated to monitor a fault on mode k [12]. The conservatism of this bound can also be demonstrated using a parity space representation, shown in Figure 3.6. Recall that the test statistics are the projections of the parity vector onto the fault mode lines. In general, if a fault is present on a particular mode, the resulting parity vector will lie close to that fault mode line. While the true no detection region is the interior of the white hexagon, the no detection region for mode k in equation 3.20 is the gray stripe labelled ND_k (including the white hexagon). This may be considered a tight bound as well because the probability of the parity vector lying in the gray regions when a fault is present on mode k is very small, as the parity vector will generally be close to the fault mode line. Thus, the integrity risk of the detection

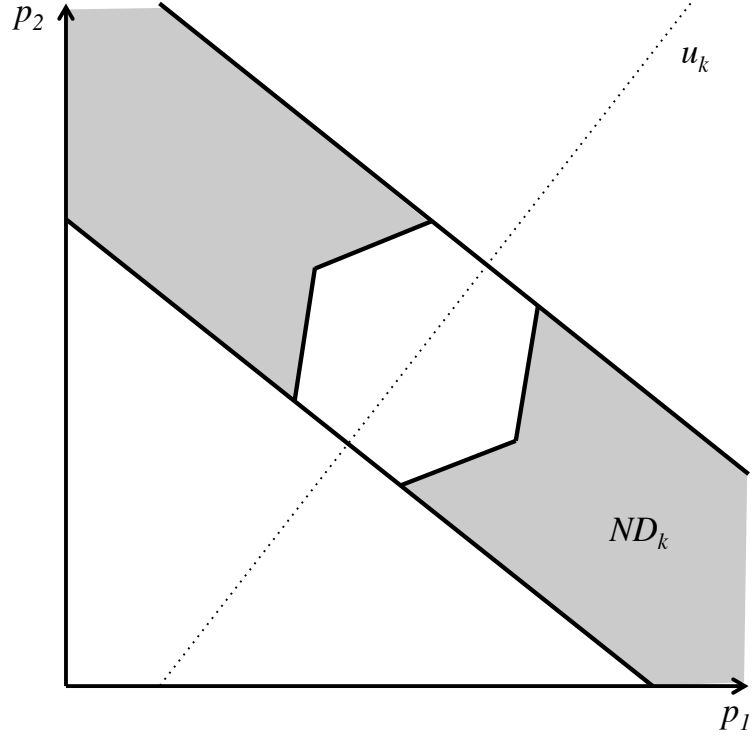


Figure 3.6. Integrity Risk Upper Bound in Parity Space.

function is upper bounded as

$$\sum_{k=0}^n P \left(|\varepsilon_0| > L \cap \left(\bigcap_{k=1}^n |q_k| < T_k \right) \middle| H_k \right) P_{H_k} \leq P(|\varepsilon_0| > L | H_0) P_{H_0} + \sum_{k=1}^n P(|\varepsilon_0| > L \cap |q_k| < T_k | H_k) P_{H_k}. \quad (3.21)$$

The second term on the right side of this equation, which corresponds to missed detection (MD), can be visualized using a fault mode plot, shown in Figure 3.7. This plot is similar to Figure 3.5, but now the covariance under fault mode k is of interest. The diagonal line is the fault mode line; as the fault magnitude varies, the center of the covariance ellipse is shifted along this line. With the detection thresholds set to limit the FANE contribution to the continuity risk for each mode, the integrity risk can be determined using those very thresholds (this will be shown later).

Now, consider the integrity risk of the exclusion function, given by the second

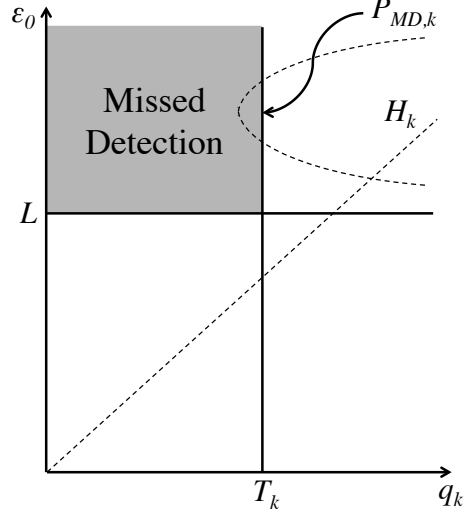


Figure 3.7. Fault Mode Plot.

term on the right side of equation 3.17. It can first be bounded by

$$\sum_{e=1}^{n_e} \sum_{j=0}^n P \left(|\varepsilon_e| > L \cap \left(\bigcap_{\substack{j=1 \\ j \neq e}}^n |q_{e,j}| < T_{e,j} \right) \cap \left(\bigcup_{k=1}^n |q_k| > T_k \right) \middle| H_j \right) P_{H_j} \leq$$

$$\sum_{e=1}^{n_e} \sum_{j=0}^n P \left(|\varepsilon_e| > L \cap \left(\bigcap_{\substack{j=1 \\ j \neq e}}^n |q_{e,j}| < T_{e,j} \right) \middle| H_j \right) P_{H_j} \quad (3.22)$$

by ignoring the knowledge of the first layer detection tests. Essentially, this bound ignores the knowledge that a fault was detected in the first place. This is a tight upper bound because, if a fault is present, the probability of detecting it is close to one. This can then be split into the correct exclusion ($j = e$) and wrong exclusion

($j \neq e$) terms as

$$\begin{aligned} \sum_{e=1}^{n_e} \sum_{j=0}^n P \left(|\varepsilon_e| > L \cap \left(\bigcap_{\substack{j=1 \\ j \neq e}}^n |q_{e,j}| < T_{e,j} \right) \middle| H_j \right) P_{H_j} = \\ \sum_{e=1}^{n_e} \left[P \left(|\varepsilon_e| > L \cap \left(\bigcap_{\substack{j=1 \\ j \neq e}}^n |q_{e,j}| < T_{e,j} \right) \middle| H_0 \right) P_{H_e} + \right. \\ \left. \sum_{\substack{j=0 \\ j \neq e}}^n P \left(|\varepsilon_e| > L \cap \left(\bigcap_{\substack{j=1 \\ j \neq e}}^n |q_{e,j}| < T_{e,j} \right) \middle| H_j \right) P_{H_j} \right]. \quad (3.23) \end{aligned}$$

The first term on the right side of the above equation can be upper bounded as

$$\sum_{e=1}^{n_e} P \left(|\varepsilon_e| > L \cap \left(\bigcap_{\substack{j=1 \\ j \neq e}}^n |q_{e,j}| < T_{e,j} \right) \middle| H_0 \right) P_{H_e} \leq \sum_{e=1}^{n_e} P(|\varepsilon_e| > L | H_0) P_{H_e}. \quad (3.24)$$

Notice that, if the fault is correctly excluded, the estimate using the remaining satellites is fault-free, and thus its distribution is described by H_0 . This is a tight bound because, if there is a fault on mode e and it is attempting to be excluded, the probability of no second layer detection after omission of mode e is close to one. Just as was done for the missed detection terms, the wrong exclusion terms for each exclusion option e can be bounded by ignoring knowledge of all exclusion test statistics except for the one specifically designated to monitor a fault on second layer mode j , yielding

$$\begin{aligned} \sum_{\substack{j=0 \\ j \neq e}}^n P \left(|\varepsilon_e| > L \cap \left(\bigcap_{\substack{j=1 \\ j \neq e}}^n |q_{e,j}| < T_{e,j} \right) \middle| H_j \right) P_{H_j} \leq \\ P(|\varepsilon_e| > L | H_0) P_{H_0} + \sum_{\substack{j=1 \\ j \neq e}}^n P(|\varepsilon_e| > L \cap |q_{e,j}| < T_{e,j} | H_j) P_{H_j}. \quad (3.25) \end{aligned}$$

This is a tight bound in the same way it was for the integrity risk due to missed detection (see equations 3.19 and 3.20). If the wrong mode is excluded, a fault will still be present in the remaining satellites. Thus, the parity vector will still lie close

to its corresponding fault mode line, and so its probability of landing in the detection regions for all other fault modes is small (see Figure 3.6). The first term in the above equation corresponds to “unnecessary” exclusion; no fault is present (P_{H_0}), but mode e is excluded.

In summary, the total integrity risk of the detection and exclusion functions is upper bounded as

$$P_{HMI} \leq P(|\varepsilon_0| > L|H_0) P_{H_0} + \sum_{k=1}^n P(|\varepsilon_0| > L \cap |q_k| < T_k|H_k) P_{H_k} + \sum_{e=1}^{n_e} \left[P(|\varepsilon_e| > L|H_0) P_{H_e} + P(|\varepsilon_e| > L|H_0) P_{H_0} + \sum_{\substack{j=1 \\ j \neq e}}^n P(|\varepsilon_e| > L \cap |q_{e,j}| < T_{e,j}|H_j) P_{H_j} \right]. \quad (3.26)$$

The five terms in this equation correspond to HMI under fault-free conditions, HMI due to missed detection, HMI after correct exclusion, HMI after unnecessary exclusion, and HMI due to wrong exclusion. It should be noted that the estimate error distributions after correct and unnecessary exclusion (the third and fourth terms on the right side of the equation) are not the same as the estimate error distribution of the first term, despite H_0 describing each. This notation is only to demonstrate that the distributions are fault-free; the distributions after correct or unnecessary exclusion are fault-free after excluding mode e , whereas the distribution of the first term is fault-free using all satellites in view.

It should also be noted that if the user is not implementing exclusion, the integrity risk terms associated with the exclusion function may be omitted. This highlights the tradeoff of the exclusion function; it trades integrity for continuity by giving the user the ability to exclude a fault if one is detected.

CHAPTER 4

ARAIM ALGORITHM

4.1 Error Model

A crucial part of the ARAIM algorithm is accurately modeling measurement errors. The measurement variances of each satellite are used to form covariance matrices that are used in position estimation, protection level calculation, and FDE. For the purposes of ARAIM, two covariance matrices are generated, one for accuracy and continuity and one for integrity. For accuracy and continuity, the user range error (URE) is used instead of the URA. In practice, the URE is defined to be

$$\sigma_{URE} = \frac{2}{3}\sigma_{URA}, \quad (4.1)$$

in [13] and is a representation of the expected instantaneous SIS error standard deviation instead of the maximum SIS error standard deviation over the navigation message interval. As a result, the pseudorange measurement error of satellite i for accuracy and continuity is given by

$$\sigma_{acc,i}^2 = \sigma_{URE,i}^2 + \sigma_{user,i}^2 + \sigma_{tropo,i}^2. \quad (4.2)$$

Recall that the ionospheric effects have been eliminated and multipath has been mitigated by carrier smoothing. Appendix C contains a simple airborne error model for GPS and Galileo that can be used to calculate the values for the user and tropospheric errors. Because no suitable ARAIM airborne error model for GLONASS exists at the time of writing, GLONASS errors are modeled using the GPS error model. For integrity purposes, the actual URA is used, as it is more conservative. This conservatism is important when considering the safety of the ARAIM algorithm. Thus, the pseudorange measurement error of satellite i for integrity is given by

$$\sigma_{int,i}^2 = \sigma_{URA,i}^2 + \sigma_{user,i}^2 + \sigma_{tropo,i}^2. \quad (4.3)$$

The diagonal covariance matrices for accuracy and continuity and for integrity are then

$$C_{acc} = \begin{bmatrix} \sigma_{acc,1}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{acc,2}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{acc,n}^2 \end{bmatrix} \quad (4.4)$$

and

$$C_{int} = \begin{bmatrix} \sigma_{int,1}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{int,2}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{int,n}^2 \end{bmatrix}, \quad (4.5)$$

respectively, where n is the total number of satellites. In practice, the accuracy and continuity covariance matrix may be used in position estimation in equation 2.23 (i.e. $W = C_{acc}^{-1}$).

4.2 Integrity Support Message

ARAIM relies heavily on knowledge of SIS reliability and performance in the form of the integrity support message (ISM). It contains information regarding the prior probabilities of satellite and constellation faults, nominal ranging biases, and URA and URE multipliers. Table 4.1 lists the contents of the ISM [5]. Typical values for the prior probabilities of satellite and constellation fault are shown in Table 4.2 for each GNSS [5]. These values are given as a probability per hour (this notation will be suppressed for convenience). The probabilities of satellite and constellation fault for GPS are based on the GPS definition of a fault, which is an instantaneous ranging error of at least 4.42 times the broadcast URA while the navigation message designates the satellite as healthy [14]. However, because neither GLONASS nor

Table 4.1. ISM Contents.

Field	Description
$P_{sat,i}$	Prior probability of fault on satellite i
$P_{const,j}$	Prior probability of fault on constellation j
$b_{nom,i}$	Maximum nominal ranging bias for satellite i
$\alpha_{URA,i}$	URA multiplier for satellite i for integrity
$\alpha_{URE,i}$	URA multiplier for satellite i for accuracy and continuity

Table 4.2. Fault Probabilities for H-ARAIM and V-ARAIM.

	H-ARAIM			V-ARAIM		
	GPS	GLO	GAL	GPS	GLO	GAL
P_{sat} / hr	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-5}
P_{const} / hr	10^{-8}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}

Galileo have made any assurances regarding SIS reliability, and because neither have provided a rigorous definition of a fault, the values of the fault probabilities for these two GNSS are target values that anticipate future performance.

One of the primary differences between H-ARAIM and V-ARAIM is the assumption on the prior probability of GPS constellation fault. For H-ARAIM, it is considered to be 10^{-8} , which, for all intents and purposes, is essentially zero. It can be assumed small because existing RAIM, which uses GPS only, has safely considered the probability of GPS constellation fault to be zero during its service history. Furthermore, for H-ARAIM operations, HMI is only considered a major failure, whereas

it is considered a hazardous failure for V-ARAIM operations, and thus the probability of GPS constellation fault must be assumed 10^{-4} [5] for safety.

It is interesting to note that the probability of constellation fault is larger than the probability of satellite fault. This is because satellite faults and constellation faults arise from different sources. Individual satellite faults are generally caused by rare orbit and clock errors, while constellation faults are due to Earth orientation parameter (EOP) faults. EOPs describe the motion and orientation of the Earth beneath the constellation of satellites. It is known that the rotation rate of Earth is not constant and that the Earth nutates about its axis of rotation. These factors can cause constellation-wide faults when the control segment uploads satellite orbit and clock information to the constellation without properly accounting for variations in these EOPs.

The maximum nominal bias in the ISM is a conservative bound on the ranging bias of the satellite; it describes the maximum absolute value of the mean of the SIS error that can be expected under normal conditions. The URA describes the standard deviation of the SIS error overbounding distribution, as discussed previously. The parameters α_{URA} and α_{URE} are URA and URE multipliers for integrity and accuracy, respectively. These multipliers are determined by the aircraft navigation service provider (ANSP) and disseminated to users of that particular service. URAs, which are used for integrity, may be inflated in situations where safety is critical and any use of optimistic values could pose a safety risk. Conversely, UREs may be deflated in situations in which confidence is high that actual SIS errors are much less than those broadcast. For example, GPS SIS accuracies are typically below one meter; however, the minimum broadcast URA is currently 2.4 meters [7]. Thus, UREs may be deflated to increase operational availability.

It is important to note that the ISM is not currently available for ARAIM

users, and thus current ARAIM performance analysis relies on potential ISM values that reflect either conservative or anticipated GNSS performance. There are many proposals as to how the ISM be disseminated to users [5]. These include a hard-coded ISM within aviation receivers and an ISM disseminated by either existing ground infrastructure or by the GNSS themselves. Furthermore, it is unknown which aviation operations may require an updatable ISM, and which may only require a static ISM. An updatable ISM may be updated infrequently in reality, but its capability to reflect current GNSS performance and to incorporate new GNSS upon FOC may be crucial.

4.3 Protection Levels

Protection levels “protect” the user by quantifying how much confidence can be placed in the position estimate generated by the navigation system. They are set in order to satisfy the integrity risk requirement and form a cylinder around the aircraft position estimate, as shown in Figure 4.1, where the horizontal protection level (HPL) describes the radius of the cylinder and the vertical protection level (VPL) describes the height. The probability that the actual user position is outside

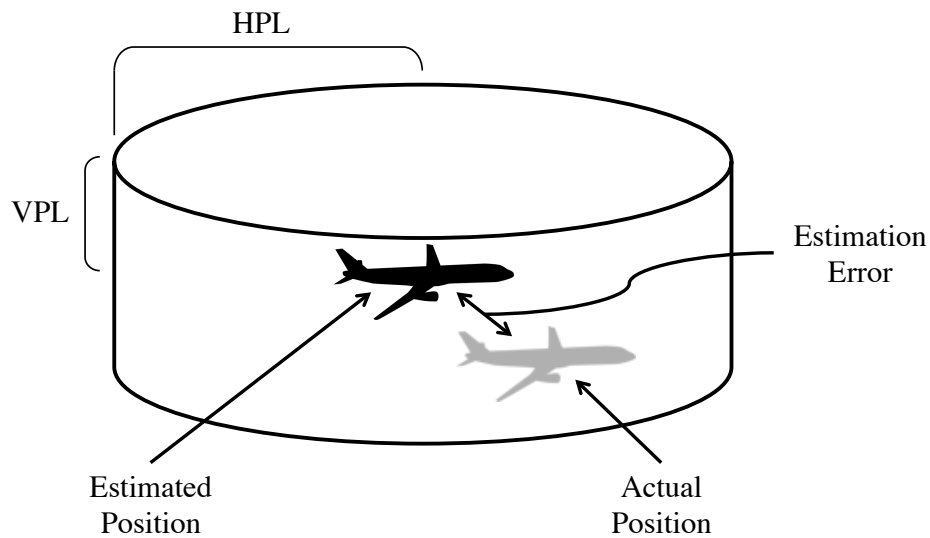


Figure 4.1. Physical Interpretation of Protection Levels.

this cylinder (i.e. the estimation error exceeds the protection level) is no larger than the integrity risk requirement. Essentially, protection levels tell the user whether or not satellite geometry is strong enough to allow for the execution of the detection and exclusion functions. Protection level computation involves determining which fault modes must be monitored and computing the detection and exclusion thresholds for those modes in order to satisfy the continuity risk requirement. These steps will be explained next.

4.3.1 Monitored Fault Modes. The user determines which fault modes to monitor based on the contents of the ISM, namely P_{sat} and P_{const} , and the unmonitored probability requirement P_{thres} . This requirement limits the probability of unmonitored faults occurring, P_{NM} , such that

$$P_{NM} \leq P_{thres}, \quad (4.6)$$

and must be smaller than the integrity risk requirement. In this work, a requirement of $P_{thres} = 8 \times 10^{-8}$ is used [5]. In order to satisfy the requirement by monitoring the fewest number of modes, those with relatively large probabilities of occurrence are monitored first, i.e. single-satellite and single-constellation fault modes. Figure 4.2 shows a typical fault mode matrix for H-ARAIM using two constellations, GPS and Galileo, whose fault probabilities can be found in Table 4.2. A gray cell indicates that the satellite is hypothesized to be faulted.

For H-ARAIM, monitoring single-fault modes is generally sufficient because the probability of GPS constellation fault can be assumed to be very low (essentially zero). In fact, the GPS constellation fault mode does not need to be monitored at all for H-ARAIM; this is why GPS is the only constellation capable of supporting horizontal guidance without the aid of other constellations. It is important to note that, in order to monitor a mode, it must have a sufficient number of satellites hypothesized to be fault-free in order to allow for position estimation using those fault-free satellites

Mode	GPS					Galileo				
	SV 1	SV 2	SV 3	SV 4	SV 5	SV 1	SV 2	SV 3	SV 4	SV 5
Fault-Free										
Single-SV Faults										
Galileo Constellation Fault										

Figure 4.2. Fault Mode Matrix.

(see Table 2.1). For example, if there are eight GPS satellites and three Galileo satellites in view, the GPS constellation fault mode cannot be monitored because three Galileo satellites are not sufficient to compute a position estimate. Table 4.3 shows the number of satellites required for detection for H-ARAIM. Here, the removal of any one satellite, the entire GLONASS constellation, or the entire Galileo constellation

Table 4.3. Number of Satellites Required for Detection for H-ARAIM.

Constellations	Number of Satellites
GPS	5
GPS + GLO/GAL	4 + 2
GPS + GLO + GAL	3 + 2 + 2

still leaves enough satellites remaining in order to compute a position estimate.

The number of satellites required for detection is different for V-ARAIM because the probability of GPS constellation fault cannot be assumed small. Thus, the GPS constellation fault mode must be monitored, and therefore more satellites are required from other constellations. Table 4.4 shows the number of satellites required for detection for V-ARAIM. Now, there must be sufficient redundancy to compute a

Table 4.4. Number of Satellites Required for Detection for V-ARAIM.

Number of Constellations	Number of Satellites
2	4 + 4
3	3 + 3 + 2

position estimate after the removal of any one satellite or any one constellation. As such, V-ARAIM always requires measurements from multiple constellations to allow for the removal of any one of them. It is important to note that, because the probability of GPS constellation fault is large for V-ARAIM, the user may be required to monitor multiple-fault modes, such as dual-satellite or satellite/constellation modes, in order to satisfy the unmonitored probability requirement. In these cases, more

satellites may be required in order to perform detection.

In any case, the associated fault mode probabilities P_{H_k} are calculated for each mode k by

$$P_{H_k} = \prod_{j=1}^{n_{const}} \left[\left((P_{sat,j})^{n_{F,j}} (1 - P_{sat,j})^{n_{sat,j} - n_{F,j}} (1 - P_{const,j}) \right) (1 - f_j) + P_{const,j} f_j \right], \quad (4.7)$$

including the fault-free mode H_0 , whose probability is very close to one. The total number of satellites in constellation j is denoted as $n_{sat,j}$, and the number of satellites hypothesized to be faulted in constellation j is denoted as $n_{F,j}$. The value f_j is equal to one if constellation j is hypothesized to be faulted and zero if it is not. Note that if a constellation is hypothesized to be faulted, any individual satellites within that constellation are ignored (i.e. they are not assumed either faulted or fault-free). In this equation, the assumption is made that the probabilities of satellite fault are the same for all satellites within an individual constellation. It is also important to note that the monitored modes are mutually exclusive, and thus the total monitored probability must be less than one (since not all fault modes are monitored). This is shown by

$$\sum_{k=0}^n P_{H_k} + P_{NM} = 1, \quad (4.8)$$

where n is the total number of monitored modes. If $P_{NM} > P_{thres}$, then more fault modes (satellite/constellation modes, dual-satellite modes, etc.) must be monitored.

4.3.2 Detection Thresholds. The continuity requirement allocated to false alarm, P_{FANE} , is used to compute the detection thresholds. Because the upper bound on the continuity risk due to false alarm (the first term on the right side of equation 3.16) implies the mutual independence of the different modes, the FANE budget is distributed among them. Thus, the detection threshold for mode k is set to limit the

false alarm probability of that mode, as

$$P(|q_k| > T_k | H_0) = \frac{P_{FANE}}{P_{H_0} n}. \quad (4.9)$$

Because we wish to calculate protection levels in each direction, there are three test statistics and thresholds for each mode, one for each direction d . The FANE requirement may then be distributed between the horizontal and vertical directions as $P_{FANE,hor}$ and $P_{FANE,vert}$, respectively. The thresholds for the horizontal directions, $d = 1, 2$, are then defined as

$$T_{k,d} = Q^{-1} \left(\frac{P_{FANE,hor}}{4P_{H_0} n} \right) \sigma_{ss,k,d} \quad (4.10)$$

and the thresholds for the vertical direction, $d = 3$, are defined as

$$T_{k,d} = Q^{-1} \left(\frac{P_{FANE,vert}}{2P_{H_0} n} \right) \sigma_{ss,k,d}, \quad (4.11)$$

where $Q^{-1}(x)$ is the inverse of the Q-function, defined as

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt, \quad (4.12)$$

which is the right tail probability of a normal distribution. In the above equations, $\sigma_{ss,k,d}$ denotes square root of the d -th diagonal element of the covariance matrix for the difference between the all-in-view position solution and the subset position solution corresponding to mode k , given by

$$C_{ss,k} = (S_k - S_0) C_{acc} (S_k - S_0)^T. \quad (4.13)$$

Recall the definition of S , which will subsequently be referred to as S_0 to indicate the all-in-view set of satellites, in equation 2.24. S_k is calculated as

$$S_k = (G^T W_k G)^{-1} G^T W_k, \quad (4.14)$$

where W_k is the same as the weighting matrix W , but with the entries corresponding to the satellite(s) hypothesized to be faulted in mode k set to zero. In practice, the

fault-free probability P_{H_0} in the denominators of the arguments of the Q-functions in equations 4.10 and 4.11 may be bounded by one for convenience. The factors of four and two in the denominators of the arguments of the Q-functions divide $P_{FANE,hor}$ into the four horizontal directions (North, South, East, and West) and $P_{FANE,vert}$ into the two vertical directions (up and down) because of the two-tailed probability in equation 4.9.

4.3.3 Exclusion Thresholds. When implementing the exclusion function, the user must account for the contribution of the exclusion function to the integrity risk, and thus must compute exclusion thresholds for all possible exclusion options. It is important to note that not all monitored fault modes may be excluded, because in order to exclude, there must be sufficient redundancy in order to perform second layer detection on the remaining satellites. Table 4.5 shows the required number of satellites in order to be able to implement the exclusion function for H-ARAIM. When using one

Table 4.5. Number of Satellites Required for Exclusion for H-ARAIM.

Constellations	Number of Satellites
GPS	6
GPS + GLO/GAL	5 + 4
GPS + GLO + GAL	4 + 4 + 4

constellation, there must be sufficient GPS satellite redundancy to monitor all GPS satellite faults after the exclusion of any one satellite. When using two constellations, there must be sufficient GPS satellite redundancy that the exclusion of any GPS satellite still enables the (necessary) monitoring of either the GLONASS or Galileo constellation fault mode. There must also be sufficient GPS satellite redundancy that the exclusion of the GLONASS or Galileo constellation still enables the monitoring

of all GPS satellite fault modes. Finally, when using three constellations, having four satellites from each is sufficient to allow the monitoring of the GLONASS or Galileo constellation fault modes even after an entire constellation has been omitted.

For V-ARAIM, in which all constellations perform equally, the numbers of required satellites for exclusion are listed in Table 4.6. It is important to note that,

Table 4.6. Number of Satellites Required for Exclusion for V-ARAIM.

Number of Constellations	Number of Satellites
2	5 + 5
3	4 + 4 + 4

when using two constellations, satellite faults may be excluded, but neither constellation may be excluded. This is because detection cannot be performed on the remaining constellation once one is omitted, since there is now no constellation redundancy. However, when using three full constellations, any one constellation may be excluded because detection can be performed on the two remaining constellations, as the omission of either one still leaves an entire constellation to evaluate a position estimate.

The exclusion thresholds are set in much the same way as the detection thresholds, only using the FDNE requirement P_{FDNE} . Again, the user must account for all possible exclusion options in order to properly characterize the integrity risk associated with the exclusion function. Recall the upper bound on the continuity risk due to FDNE, the second term on the right side of equation 3.16. The exclusion thresholds are set such that

$$P(|q_{e,j}| > T_{e,j} | H_0) = \frac{P_{FDNE}}{P_{He} n(n-1)}, \quad (4.15)$$

where n is the number of first layer detection modes, $n - 1$ is the number of second layer detection modes, and e indexes the mode attempting to be excluded. Here, the total FDNE requirement is evenly allocated among the second layer modes, since the user must consider all possible exclusion options. Note that exclusion thresholds are only computed for modes that can be excluded, because not all monitored modes k are exclusion options e . For example, for dual-constellation V-ARAIM, both constellation modes must be monitored, but neither may be excluded (for reasons discussed previously); as such, both constellation fault modes are in $k = 1, 2, \dots, n$, but neither is in $e = 1, 2, \dots, n_e$, so exclusion thresholds cannot be computed for those modes. However, the FDNE budget must still be allocated to those modes that cannot be excluded because their occurrence (and detection) poses a continuity threat, hence the n in the denominator of the right side of equation 4.15 and not n_e .

Again, because we wish to compute protection levels in each direction, the FDNE requirement may be allocated to the horizontal and vertical directions as $P_{FDNE,hor}$ and $P_{FDNE,vert}$, respectively. Thus, the exclusion thresholds in direction d for second layer mode j after attempting to exclude mode e are then calculated as

$$T_{e,j,d} = Q^{-1} \left(\frac{P_{FDNE,hor}}{4P_{H_e} n (n - 1)} \right) \sigma_{ss,e,j,d} \quad (4.16)$$

in the horizontal directions and

$$T_{e,j,d} = Q^{-1} \left(\frac{P_{FDNE,vert}}{2P_{H_e} n (n - 1)} \right) \sigma_{ss,e,j,d} \quad (4.17)$$

in the vertical direction. In the above equations, $\sigma_{ss,e,j,d}$ denotes square root of the d -th diagonal element of the covariance matrix for the difference between the all-in-view position solution after omission of mode e and the subset position solution corresponding to mode j after omission of mode e , given by

$$C_{ss,e,j} = (S_{e,j} - S_e) C_{acc} (S_{e,j} - S_e)^T, \quad (4.18)$$

where S_e can be determined from equation 4.14 and $S_{e,j}$ is calculated as

$$S_{e,j} = (G^T W_{e,j} G)^{-1} G^T W_{e,j}. \quad (4.19)$$

Now, $W_{e,j}$ is the same as W , but with the entries corresponding to the satellite(s) in mode e attempting to be excluded and the entries corresponding to the satellite(s) hypothesized to be faulted in second layer mode j set to zero.

4.3.4 Protection Level Calculation. Recall that the total integrity risk is upper-bounded by equation 3.26. Now, the arbitrary limit L has been replaced by the protection level PL . Because the protection levels are set to guarantee that the integrity risk requirement is met, P_{HMI} is replaced by the integrity risk requirement I_{req} , from which the unmonitored probability is subtracted. Thus, the protection level equation for detection and exclusion is

$$I_{req} - P_{NM} = P(|\varepsilon_0| > PL | H_0) P_{H_0} + \sum_{k=1}^n P(|\varepsilon_0| > PL \cap |q_k| < T_k | H_k) P_{H_k} + \sum_{e=1}^{n_e} \left[P(|\varepsilon_e| > PL | H_0) P_{H_0} + \sum_{\substack{j=1 \\ j \neq e}}^n P(|\varepsilon_e| > PL \cap |q_{e,j}| < T_{e,j} | H_j) P_{H_j} \right]. \quad (4.20)$$

Because determining the protection levels from this equation is difficult, it is upper bounded as

$$I_{req} - P_{NM} \leq 2Q \left(\frac{PL - b_0}{\sigma_0} \right) + \sum_{k=1}^n P_{H_k} Q \left(\frac{PL - b_k - T_k}{\sigma_k} \right) + \sum_{e=1}^{n_e} \left[2Q \left(\frac{PL - b_e}{\sigma_e} \right) + \sum_{\substack{j=1 \\ j \neq e}}^n P_{H_j} Q \left(\frac{PL - b_{e,j} - T_{e,j}}{\sigma_{e,j}} \right) \right] \quad (4.21)$$

in which all terms are one-sided tail probabilities. Appendix D provides the derivation of equation 4.21 from equation 4.20. As mentioned previously, three protection levels

are calculated, one for each direction d , as

$$I_{req,d} \left(1 - \frac{P_{NM}}{I_{req}} \right) \leq 2Q \left(\frac{PL_d - b_{0,d}}{\sigma_{0,d}} \right) + \sum_{k=1}^n P_{H_k} Q \left(\frac{PL_d - b_{k,d} - T_{k,d}}{\sigma_{k,d}} \right) + \sum_{e=1}^{n_e} \left[2Q \left(\frac{PL_d - b_{e,d}}{\sigma_{e,d}} \right) + \sum_{\substack{j=1 \\ j \neq e}}^n P_{H_j} Q \left(\frac{PL_d - b_{e,j,d} - T_{e,j,d}}{\sigma_{e,j,d}} \right) \right]. \quad (4.22)$$

For the detection terms, the estimation error for each monitored mode (including the fault-free mode) is described by

$$\varepsilon_{k,d} \sim N(b_{k,d}, \sigma_{k,d}^2). \quad (4.23)$$

Here, $b_{k,d}$ is the worst-case impact of the nominal ranging biases on the position solution for mode k in direction d , calculated as

$$b_{k,d} = \sum_{i=1}^{n_{sat}} |(S_k)_{d,i}| b_{nom,i} \quad (4.24)$$

where $b_{nom,i}$ is the maximum nominal ranging bias for satellite i (from the ISM) and $(S_k)_{d,i}$ is the d, i entry of the matrix S_k . Since S is a matrix that takes values from the measurement domain and projects them into the position domain, this worst-case bias is simply the projection of all nominal ranging biases into the position domain in a way that adds all their impacts together. The variance $\sigma_{k,d}^2$ is the d -th diagonal entry of the covariance matrix for the position estimate corresponding to mode k , defined as

$$C_k = (G^T W_k G)^{-1}. \quad (4.25)$$

Recall that the detection thresholds $T_{k,d}$ are calculated according to equations 4.10 and 4.11.

For the exclusion terms, the estimation errors are distributed as

$$\varepsilon_{e,j,d} \sim N(b_{e,j,d}, \sigma_{e,j,d}^2), \quad (4.26)$$

where $b_{e,j,d}$ is calculated as

$$b_{e,j,d} = \sum_{i=1}^{n_{sat}} |(S_{e,j})_{d,i}| b_{nom,i} \quad (4.27)$$

and $\sigma_{e,j,d}^2$ is the d -th diagonal element of

$$C_{e,j} = (G^T W_{e,j} G)^{-1}. \quad (4.28)$$

Recall that the exclusion thresholds $T_{e,j,d}$ are calculated according to equations 4.16 and 4.17.

Because equation 4.22 cannot be solved for PL_d in closed form, Appendix E presents an iterative procedure to do so. The HPL, which describes the radius of the cylinder in Figure 4.1, is calculated as

$$HPL = \sqrt{HPL_E^2 + HPL_N^2}, \quad (4.29)$$

which is the root-sum-square (RSS) of the HPLs in the East/West and North/South directions.

4.4 Availability

ARAIM performance is quantified in terms of availability, which is the fraction of time that the operational requirements are met. Aviation operations can be separated into two main categories: those requiring horizontal guidance only (H-ARAIM) and those requiring both horizontal and vertical guidance (V-ARAIM). H-ARAIM operations are for phases of flight such as en route flight, terminal and departure operations, and initial approach [11]. V-ARAIM operations are those that include aircraft approach, for which vertical guidance is of utmost importance.

Different operations within the categories of H-ARAIM and V-ARAIM have different operational requirements, which are often specified in terms of maximum allowable protection levels, called alert limits (AL). Like the protection levels, the alert limits form a cylinder around the aircraft, as shown in Figure 4.3. Recall that the probability that the actual aircraft position is outside the protection levels is no greater than the integrity risk requirement. H-ARAIM operations, like RNP 0.1 and

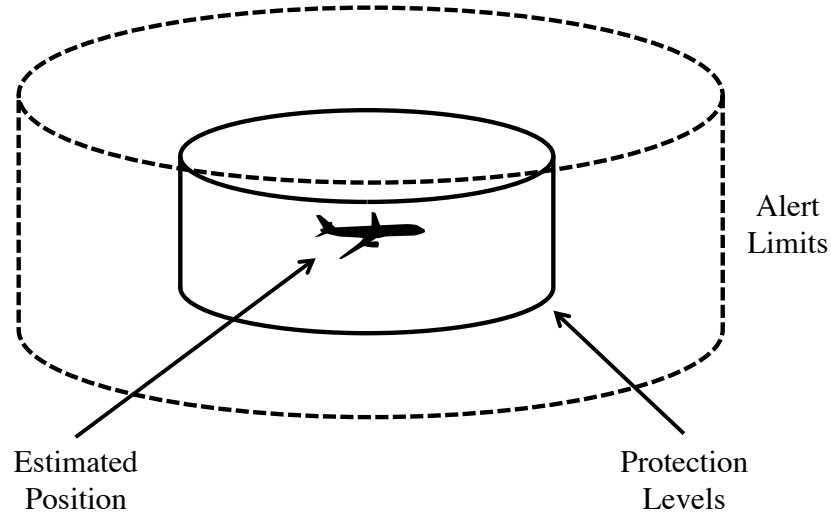


Figure 4.3. Protection Levels and Alert Limits.

RNP 0.3 (RNP stands for required navigation performance), only specify a requirement on the horizontal protection levels. On the other hand, V-ARAIM operations, such as LPV-200, which requires vertical guidance down to 200 feet altitude for aircraft approach, specify requirements on both the horizontal and vertical protection levels, as well as requirements on vertical accuracy and the effective monitor threshold (EMT). The operational requirements for RNP 0.3, RNP 0.1 [5], and LPV-200 [15] are listed in Table 4.7.

Table 4.7. RNP and LPV-200 Operational Requirements.

Operation	HAL	VAL	$\sigma_{vert,req}$	EMT_{req}
RNP 0.3	556 m	-	-	-
RNP 0.1	185 m	-	-	-
LPV-200	40 m	35 m	1.87 m	15 m

The vertical accuracy of the position estimate is

$$\sigma_{vert} = \sqrt{C_{3,3}}, \quad (4.30)$$

where $C_{3,3}$ is the third diagonal element of

$$C = S_0 C_{acc} S_0^T. \quad (4.31)$$

The EMT is simply

$$EMT = \max_{k,3|P_{H_k} > P_{EMT}} T_{k,3}, \quad (4.32)$$

which is the largest of the vertical detection thresholds for those modes whose probability of occurrence is larger than P_{EMT} . In this work, a value of $P_{EMT} = 10^{-6}$ is used so as to only consider the thresholds for single-fault modes. For example, even if a multiple-fault mode yields a very large threshold, it need not be considered because its probability of occurrence is so small. The EMT has also been considered the largest of the vertical detection thresholds for modes that monitor a fault whose individual fault probability (not the probability of mode occurrence, which accounts for the fact that the other satellites are hypothesized to be fault-free) is larger than $P_{EMT} = 10^{-5}$ in [4], but the formulation in equation 4.32 accomplishes the same goal.

A requirement on the EMT is set in order to ease the workload on the pilot during LPV-200 approach. Because the VAL allows for a vertical positioning error of up to 35 meters, actual positioning errors of this magnitude do not cause a loss of availability when regarding the VPL only. For instance, at 200 feet altitude, a vertical positioning error of up to 35 meters is tolerable by the VAL, but consideration of only the VPL means that the aircraft is considered safe if its true altitude is between 85 and 315 feet. In reality, positioning errors of this magnitude can greatly increase the workload on the pilot. The EMT requirement limits the largest vertical detection threshold to 15 meters; if any threshold is larger, then FDE is unavailable. This means that if the difference between any subset position estimate and the all-in-view

position estimate (i.e. the test statistic) is allowed to be larger than 15 meters without a fault being detected, then availability is lost.

The alert limits for en route flight are relatively lenient because the navigation system is only required to give a rough estimate of aircraft position in the lateral direction; altitude is generally measured using other instruments. However, the requirements for LPV-200 approach are strict because the navigation system must provide vertical guidance down to a 200-foot decision height, at which the pilot must either commit to landing or abort and circle around again.

If all requirements for an operation are met, then the FDE algorithm is declared available, at which point the user may then proceed to perform FDE. If the requirements are not met, then FDE is declared unavailable, at which point the user must abort the operation. The availability criteria reinforce the need for an ISM whose parameters accurately predict SIS performance and reliability. In a real-world application, it is unsafe for an airborne navigation system to assume too optimistic an ISM. The predicted integrity of the position estimates would then correspond to these optimistic parameters and would not reflect the actual SIS reliability or performance, thus yielding similarly optimistic protection levels. This may then lead the aircraft into a dangerous situation in which the navigation system declares availability, while in reality, satellite geometry and performance are insufficient. On the other hand, using too conservative an ISM would be detrimental to availability. Determining and validating appropriate ISM parameters is of the utmost importance to ARAIM, and is the subject of ongoing research.

4.5 Fault Detection

If the operational requirements are met, then the user may proceed to fault detection. Recall that the detection thresholds have already been computed according

to the FANE requirement. For each of the monitored fault modes, three test statistics are calculated, one for each direction. They are defined as

$$q_{k,d} = |\hat{x}_{0,d} - \hat{x}_{k,d}|, \quad (4.33)$$

which are simply the magnitudes of the differences between the all-in-view position solution \hat{x}_0 and the subset position solutions \hat{x}_k in each direction; this is the solution separation aspect of the MHSS methodology. The test statistics may also be conveniently represented as

$$q_{k,d} = |((S_0)_d - (S_k)_d)z|, \quad (4.34)$$

where d indicates the d -th row of the matrices S_0 and S_k . Figure 4.4 shows these test statistics in the position domain in a scenario in which there are five monitored modes, each with a corresponding test statistic. This representation of the test statistics makes the MHSS approach obvious. If one measurement is faulted, measurement

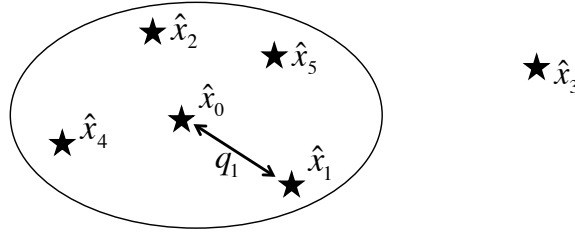


Figure 4.4. MHSS Test Statistics.

three for example, then the mode that omits that measurement is fault-free and thus may yield a significantly different position solution than the modes employing the faulted measurement (those inside the ellipse). The detection function aims to detect these cases.

If, for all modes k and directions d ,

$$q_{k,d} \leq T_{k,d}, \quad (4.35)$$

then no fault is detected. Otherwise, if any test statistic exceeds its threshold, a fault is detected and exclusion may be attempted. If the exclusion function is not implemented, then any fault detection results in a loss of continuity. As mentioned previously, the exclusion function is useful in that it allows the user to regain continuity after a fault has been detected. It is crucial to note that the faulted mode is not necessarily the one with the largest test statistic; that a test statistic exceeds its corresponding threshold only indicates that a fault is detected.

4.6 Fault Exclusion

If a fault is detected, then exclusion may be attempted in order to prevent a loss of continuity. The goal of exclusion is to exclude the detected fault and obtain a fault-free subset of satellites that can be used to compute a fault-free position estimate. In a real-time ARAIM application, exclusion can only increase continuity by giving the user the ability to exclude a detected fault and continue the operation, but at the cost of degrading integrity due to the possibility of wrong exclusion. If a fault is detected and not excluded, the user knows that a fault is present and that the position estimate generated by the navigation system cannot be trusted. However, if a fault is detected and the wrong mode is excluded, the user will then be fooled into trusting the position estimate after exclusion despite the fact that a fault remains undetected among the satellites that the user believes to be fault-free.

The first step in exclusion is determining which fault mode to attempt to exclude first. In practice, the user may decide to attempt to exclude the monitored mode with the largest normalized test statistics (i.e. $\frac{q_{k,d}}{\sigma_{ss,k,d}}$) first. This is a good place to start, as single-satellite faults generally manifest themselves as large normalized test statistics for their respective modes. However, multiple-satellite faults and constellation faults may not manifest themselves in this manner. Going back to the parity space example, a parity vector corresponding to multiple simultaneous

faults may not lie near a single-fault fault mode line, and thus will not necessarily yield a test statistic that clearly indicates on which mode the fault is. In any case, attempting to exclude the mode with the largest normalized test statistics first is often reasonable.

Once a mode has been selected to attempt to exclude (this is called the exclusion option), second layer detection is performed on the remaining subset of satellites after the satellite(s) hypothesized to be faulted in the exclusion option is/are omitted. As an example, recall the fault mode matrix shown in Figure 4.2 and suppose a fault has been detected among the single-satellite and constellation fault modes monitored. Also suppose that the largest normalized test statistics correspond to the mode that is monitoring a fault on Galileo SV 2, in which case the user decides to attempt to exclude Galileo SV 2. Therefore, Galileo SV 2 is omitted from the set of ten satellites, and second layer detection is performed on the remaining nine satellites, as shown in Figure 4.5. Here, the black cells indicate that the satellite has been omitted.

It is important to note that a mode is only an exclusion option if second layer detection is able to monitor the same fault modes after omission of the exclusion option as before. Of course, second layer detection is not required to monitor the mode that was omitted. If a constellation is omitted, second layer detection also does not need to monitor the satellite fault modes within that constellation. As such, in this H-ARAIM scenario, the Galileo constellation may be excluded, because second layer detection is not required to monitor the GPS constellation fault mode (just as first layer detection was not required to monitor it). However, if this were a V-ARAIM scenario, neither constellation would be an exclusion option because omission of either one would leave no constellation redundancy remaining, and thus neither remaining constellation fault mode could be monitored.

The second layer test statistics are calculated in much the same manner as

Mode	GPS					Galileo				
	SV 1	SV 2	SV 3	SV 4	SV 5	SV 1	SV 2	SV 3	SV 4	SV 5
Fault-Free										
Single-SV Faults										
Galileo Constellation Fault										

Figure 4.5. Fault Mode Matrix for Exclusion.

for detection. If mode e is attempting to be excluded, then the test statistics are calculated as

$$q_{e,j,d} = |\hat{x}_{e,0,d} - \hat{x}_{e,j,d}|, \quad (4.36)$$

where $\hat{x}_{e,0,d}$ indicates the all-in-view estimate after omission of mode e and $\hat{x}_{e,j,d}$ indicates the estimate after omission of mode e and second layer mode j . In this example, $\hat{x}_{e,0,d}$ is calculated by omitting Galileo SV 2, and the subset solutions $\hat{x}_{e,j,d}$ are calculated by omitting the satellite(s) hypothesized to be faulted in second layer

mode j , in addition to Galileo SV 2.

Just as before, if all

$$q_{e,j,d} \leq T_{e,j,d}, \quad (4.37)$$

for a particular exclusion option e , then no fault is detected after exclusion. In this case, it may be concluded that the faulted mode is correctly excluded, and the remaining satellites are fault-free. If, however, a fault is still detected, then the user must attempt to exclude a different mode. This process continues until a fault-free subset of satellites is found, in which case exclusion is successful, or until all exclusion options are exhausted and no fault-free subset is found, in which case the fault cannot be excluded and continuity is lost.

Recall from the methodology chapter that the exclusion function degrades integrity. It is worth a brief but important comment here that, in a real-time ARAIM implementation, the incorporation of the exclusion function does not increase the integrity risk in that it makes the navigation system less safe. Remember that protection levels are set to satisfy the integrity risk requirement, whether or not the exclusion function is implemented. As such, the exclusion function affects integrity by increasing protection levels, because they must account for all possible scenarios (correct exclusion, unnecessary exclusion, and wrong exclusion) for each exclusion option.

CHAPTER 5
RESULTS

In this work, a MHSS ARAIM prototype was built using Matlab software. This prototype consists of a custom Matlab interface with a NovAtel GNSS receiver and a Matlab graphical user interface displaying availability and performance in real-time. Essentially, the prototype executes the flow chart in Figure 3.1 in real-time. The prototype employs dual-frequency, carrier-smoothed pseudoranges from GPS, GLO-NASS, and Galileo. Preliminary, conservative performance of the baseline MHSS algorithm is examined, but the results of this work are not meant to give any indication of either good or poor performance under current GNSS conditions, nor to indicate future ARAIM capabilities. However, the results do give insight into important aspects of multi-constellation ARAIM that can be exploited in future applications. It should be noted that all protection levels presented reflect detection only. In all simulations, the continuity and integrity budgets listed in Table 5.1 are used [5] [15]. Notice that the majority of both the continuity and integrity budgets are allocated

Table 5.1. Continuity and Integrity Budget Allocations for V-ARAIM and H-ARAIM.

	V-ARAIM	H-ARAIM
$P_{FANE,vert}$	$3.9 \times 10^{-6} / 15 \text{ sec}$	0
$P_{FANE,hor}$	$9.0 \times 10^{-8} / 15 \text{ sec}$	$3.99 \times 10^{-6} / \text{hr}$
$I_{req,vert}$	$9.8 \times 10^{-8} / \text{approach (150 sec)}$	0
$I_{req,hor}$	$2.0 \times 10^{-9} / \text{approach (150 sec)}$	$1.0 \times 10^{-7} / \text{hr}$

to the vertical directions for V-ARAIM; this is because the accuracy of the vertical

component of the position estimate is generally the poorest, as satellite geometry is weaker in the vertical direction than in the horizontal direction. Because the satellite and constellation fault probabilities are specified per hour, the V-ARAIM budgets must be converted to a per-hour basis.

5.1 GPS Versus GPS + GLONASS for H-ARAIM

As mentioned previously, GPS-only RAIM has supported horizontal guidance for decades. Here, H-ARAIM performance is examined (recall that ARAIM uses dual-frequency measurements, whereas RAIM is L1 only). Using the ISM parameters for GPS and GLONASS listed in Table 5.2 [5], Figure 5.1 demonstrates the performance improvement over a sample 24-hour period by adding the GLONASS constellation. Note that the final two fields in Table 5.2 are URA and URE multipliers, respectively, not the URAs and UREs themselves; the broadcast URAs (and corresponding UREs) are used.

Table 5.2. Nominal ISM Parameters for H-ARAIM.

	GPS	GLO	GAL
P_{sat}	10^{-5}	10^{-5}	10^{-5}
P_{const}	10^{-8}	10^{-4}	10^{-4}
b_{nom}	0.75 m	0.75 m	0.75 m
α_{URA}	1	1	1
α_{URE}	1	1	1

In this sample period, the average numbers of visible GPS and GLONASS satellites are ten and six, respectively. Adding satellites improves geometry, and if

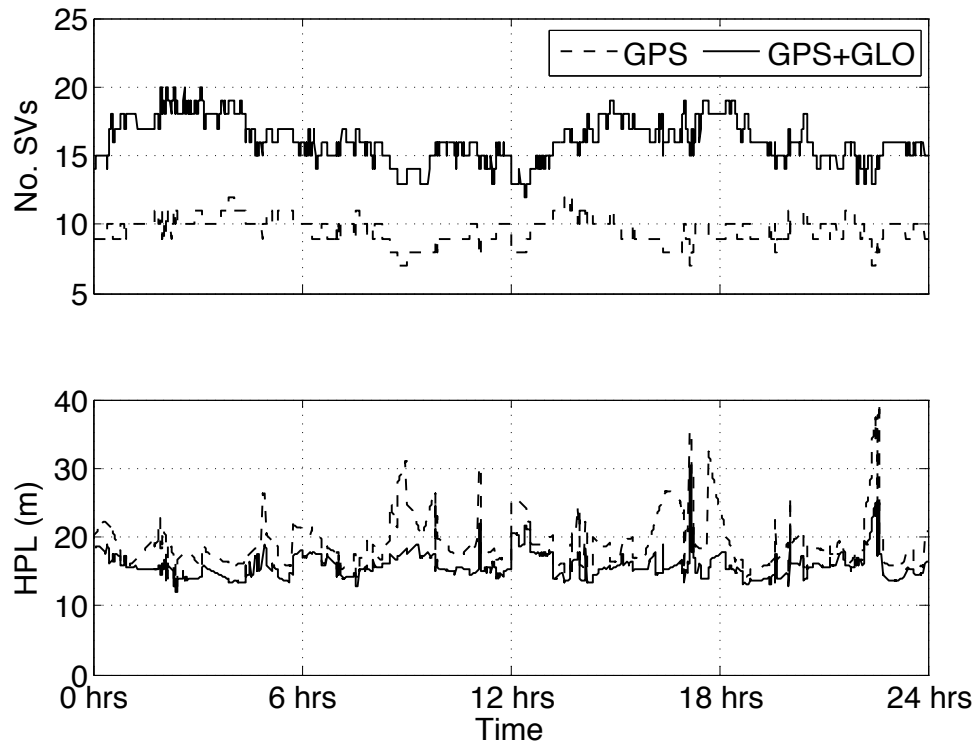


Figure 5.1. H-ARAIM Performance using GPS Only Versus GPS + GLONASS.

the additional constellation has SIS performance comparable to GPS, as it does in this case, then protection levels are improved.

5.2 H-ARAIM Versus V-ARAIM using GPS + GLONASS + Galileo

The two main differences between H-ARAIM and V-ARAIM, besides the operational requirements, are the continuity and integrity budget allocations and the prior probability of GPS constellation fault. The nominal ISM parameters for V-ARAIM are listed in Table 5.3. Because the GPS constellation fault probability must now be considered large, V-ARAIM performance is poorer than H-ARAIM performance when using multiple constellations. Figure 5.2 shows HPLs for H-ARAIM using the ISM in Table 5.2, and Figure 5.3 shows HPLs, VPLs, EMTs, and σ_{vert} for V-ARAIM using the ISM in Table 5.3. Both simulations employ GPS, GLONASS, and Galileo.

Table 5.3. Nominal ISM Parameters for V-ARAIM.

	GPS	GLO	GAL
P_{sat}	10^{-5}	10^{-5}	10^{-5}
P_{const}	10^{-4}	10^{-4}	10^{-4}
b_{nom}	0.75 m	0.75 m	0.75 m
α_{URA}	1	1	1
α_{URE}	1	1	1

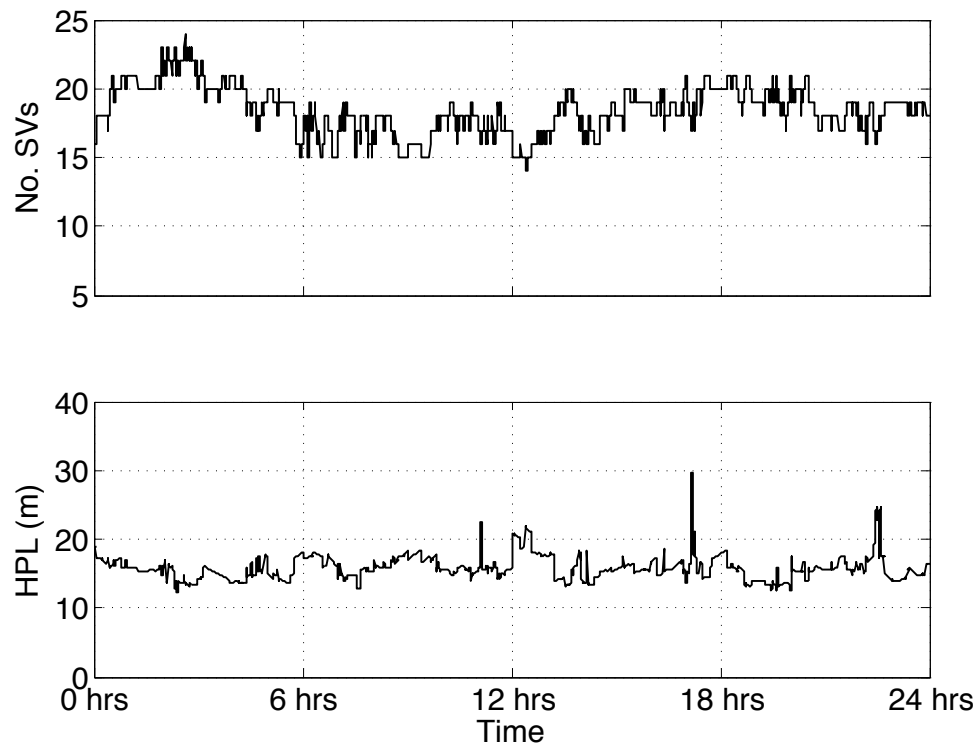


Figure 5.2. H-ARAIM Performance using GPS + GLONASS + Galileo.

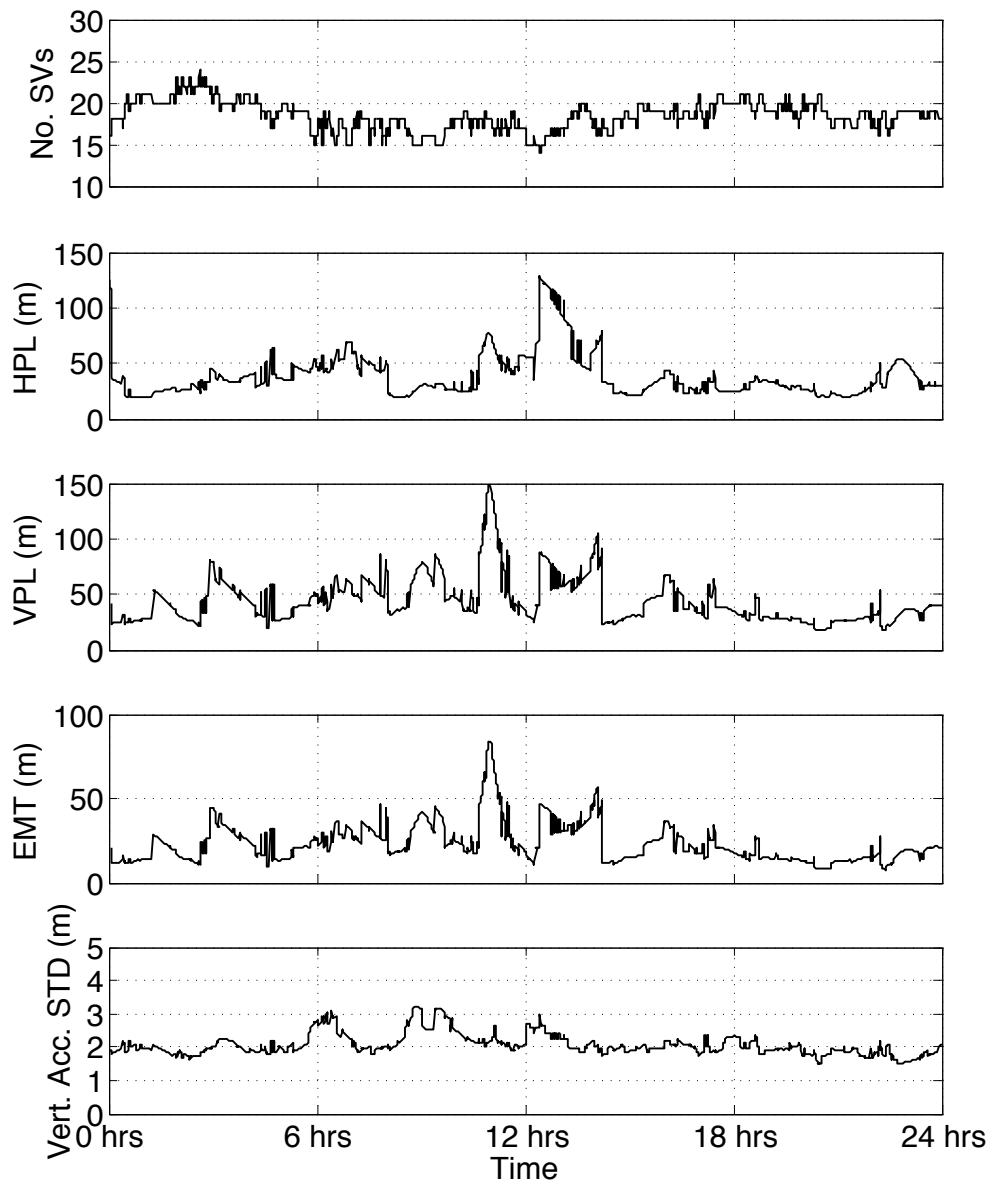


Figure 5.3. V-ARAIM Performance using GPS + GLONASS + Galileo.

This degradation in performance can be attributed to the relatively high probability of GPS constellation fault for V-ARAIM. In essence, the GPS constellation fault mode drives V-ARAIM performance. Just like all other modes, the GPS constellation fault mode contributes to the overall integrity risk of the detection function.

Its contribution to the detection terms in equation 4.21 is

$$P_{HMI,GPS} = P_{H_{GPS}} Q \left(\frac{PL - b_{GPS} - T_{GPS}}{\sigma_{GPS}} \right), \quad (5.1)$$

where the GPS subscripts on the right side of the equation indicate the GPS constellation fault mode (i.e. $k = GPS$), and the GPS subscript on the left side of the equation indicates the contribution of the GPS constellation fault mode to the overall integrity risk. We will examine three parts of this term: T_{GPS} , σ_{GPS} , and $P_{H_{GPS}}$.

Recall that the detection thresholds T_{GPS} are set to limit the false alarm probability from the GPS constellation fault mode and are computed according to equations 4.10 and 4.11. The argument of the Q-function is the same as for all other fault modes, since the FANE budget is evenly allocated among them. However, the variance of the difference between the all-in-view and subset position solutions, $\sigma_{ss,GPS}$, may be very large for the GPS constellation fault mode since all GPS satellites are omitted from subset position solution. This large variance means that the detection thresholds for the GPS constellation fault mode are large.

Next, consider the covariance of the subset solution given in equation 4.25. For the GPS constellation fault mode, this is the covariance of the position estimate using GLONASS and Galileo only, since all GPS satellites are omitted from this mode. This covariance may be very large, since omitting all GPS satellites means omitting measurements with relatively good SIS performance and dramatically degrading satellite geometry. Furthermore, both GLONASS and Galileo broadcast URAs are almost always larger than 2.4 meters, which is the most common GPS broadcast URA. Thus, the accuracy of the position estimate using GLONASS and Galileo, σ_{GPS} , is generally poor.

Finally, consider $P_{H_{GPS}}$. For V-ARAIM, the GPS constellation fault mode has a relatively high probability of occurrence. Because the integrity risk contribution of

the GPS constellation fault mode is conditioned on its prior probability of occurrence, it has a large impact on V-ARAIM performance.

The dominating nature of the GPS constellation fault mode also limits V-ARAIM performance via the EMT. As mentioned above, the GPS constellation fault mode produces large detection thresholds; the maximum vertical detection threshold, which is the EMT, often corresponds to the GPS constellation fault mode and is often larger than the requirement, as shown in Figure 5.3.

Finally, note the poor vertical positioning accuracy for V-ARAIM. In this simulation, it is almost always larger than the vertical accuracy requirement. This poor accuracy is due to the relatively large broadcast URAs. For GPS and Galileo especially, broadcast URAs/SISAs (SISA - Galileo signal-in-space accuracy; analogous to the URA) are very conservative, and thus using them has an adverse effect on availability performance. It is anticipated that future ARAIM will be able to use URA values of one meter; this will greatly improve the vertical accuracy of the position estimate, as well as the protection levels themselves.

In order to emphasize the impact of the prior probability of GPS constellation fault on performance, Figure 5.4 shows the degradation in performance for H-ARAIM only. Here, the only difference between the two curves is the prior probability of GPS constellation fault. In both cases, three constellations are used, and the entire continuity and integrity budgets are allocated to the horizontal direction. Essentially, this simulation compares performance between H-ARAIM with $P_{const, GPS} = 10^{-8}$ and H-ARAIM with $P_{const, GPS} = 10^{-4}$ using the nominal parameters for the remaining ISM contents. This is simply an exercise to show that the GPS constellation fault mode drives performance; in practice, $P_{const, GPS} = 10^{-4}$ is not used for H-ARAIM.

This result clearly demonstrates that monitoring the GPS constellation fault

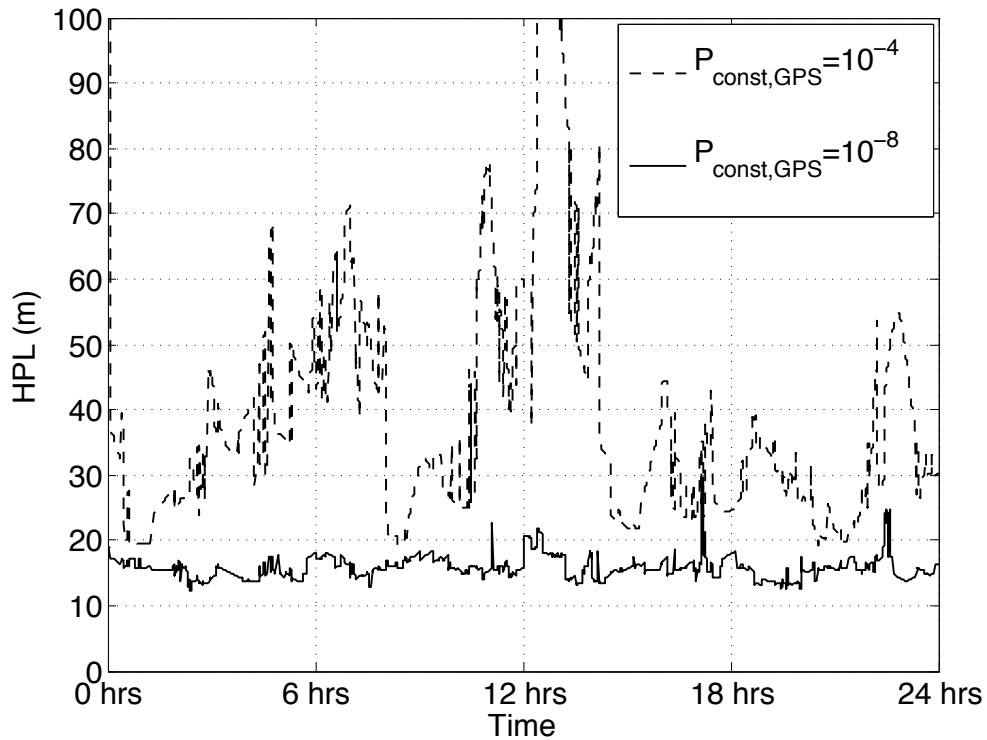


Figure 5.4. Effect of Increasing $P_{const,GPS}$ on H-ARAIM HPLs.

mode has a significant impact on protection levels. It is important to note that V-ARAIM is only supported when there are at least two full constellations. This is because, in order to satisfy the unmonitored probability requirement P_{thres} , the GPS constellation fault mode must be monitored, and it can only be monitored if there is another constellation with which to evaluate a position estimate. That the GPS constellation fault mode drives performance is an important result of this analysis, and will be used later on when considering methods of improving the computational efficiency of the ARAIM algorithm.

5.3 V-ARAIM with Degraded GLONASS Performance

In the previous two sections, all GNSS were assumed to perform at a nominal

level, with ISM parameters listed in Table 5.2 for H-ARAIM and Table 5.3 for V-ARAIM. This, however, represents anticipated performance, especially for GLONASS and Galileo, neither of whose integrities have yet been assured by a constellation service provider (CSP). GPS performance is assured and specified in a publicly available performance standard, the GPS Standard Positioning Service Performance Standard (GPS SPS PS) [14], but neither GLONASS nor Galileo have such a standard as yet. As a result, this work considers measures to more conservatively represent current SIS performance; Table 5.4 lists these more conservative ISM parameters.

Table 5.4. Degraded ISM Parameters for V-ARAIM.

	GPS	GLO	GAL
P_{sat}	10^{-5}	10^{-4}	10^{-5}
P_{const}	10^{-4}	10^{-4}	10^{-4}
b_{nom}	0.75 m	4.0 m	0.75 m
α_{URA}	1	2.4	1
α_{URE}	1	2.4	1

First, note that no degradation in Galileo SIS performance or reliability has been implemented. This is because there has been no evidence suggesting nominal biases are any larger than 0.75 meters [16]. Furthermore, just as with GPS, the Galileo broadcast SISAs are very conservative bounds on the actual SIS error. Finally, despite the fact that Galileo has experienced multiple fault events in its history, the nominal fault probabilities are appropriate.

GLONASS performance, on the other hand, has been dramatically degraded. It has been suggested in [17] that GLONASS nominal ranging biases may be much

larger than the 0.75 meter value that conservatively bounds the GPS ranging bias. As such, a nominal ranging bias of four meters is deemed suitable in this work based on the results in [17]. Furthermore, GLONASS broadcast URAs are inflated by a factor of 2.4. This is because GLONASS does not provide any assurances as to with what probability the broadcast URA bounds the SIS error. On top of this, receiver independent exchange (RINEX) navigation message files do not contain GLONASS URAs, and thus the broadcast URAs cannot be validated using this resource, which is made publicly available by the International GNSS Service (IGS). For these reasons, the minimum GLONASS broadcast URA of one meter may be optimistic, and thus URAs have been inflated such that they are never lower than the minimum GPS broadcast URA of 2.4 meters. Finally, GLONASS has been highly susceptible to faults over its service history [18]. As a result, a satellite fault probability of $P_{sat,GLO} = 10^{-4}$ is used. Furthermore, it has been suggested that the constellation fault probability may be as high as $P_{const,GLO} = 10^{-3}$ [19]. However, ARAIM cannot be implemented using $P_{const,GLO} = 10^{-3}$ with only two full constellations because the probability of both GPS and GLONASS constellations being faulted simultaneously is on the order of 10^{-7} , which is larger than P_{thres} . Because this fault mode cannot be monitored, the unmonitored probability requirement cannot be satisfied. For this reason, $P_{const,GLO}$ is set at 10^{-4} out of necessity.

Figure 5.5 compares V-ARAIM performance using the nominal ISM parameters in Table 5.3 and the degraded ISM parameters in Table 5.4, both using GPS, GLONASS, and Galileo. The results are predictable - VPLs, HPLs, EMTs, and vertical accuracy are all degraded. Inflated URAs degrade the accuracies of the position estimates for the monitored modes, σ_k , including the accuracy of the fault-free mode. This causes protection levels to increase and vertical accuracy to decrease. EMTs are also increased since the variance of the difference between the all-in-view position solution σ_0 and the subset position solution σ_k is increased due to inflated URAs.

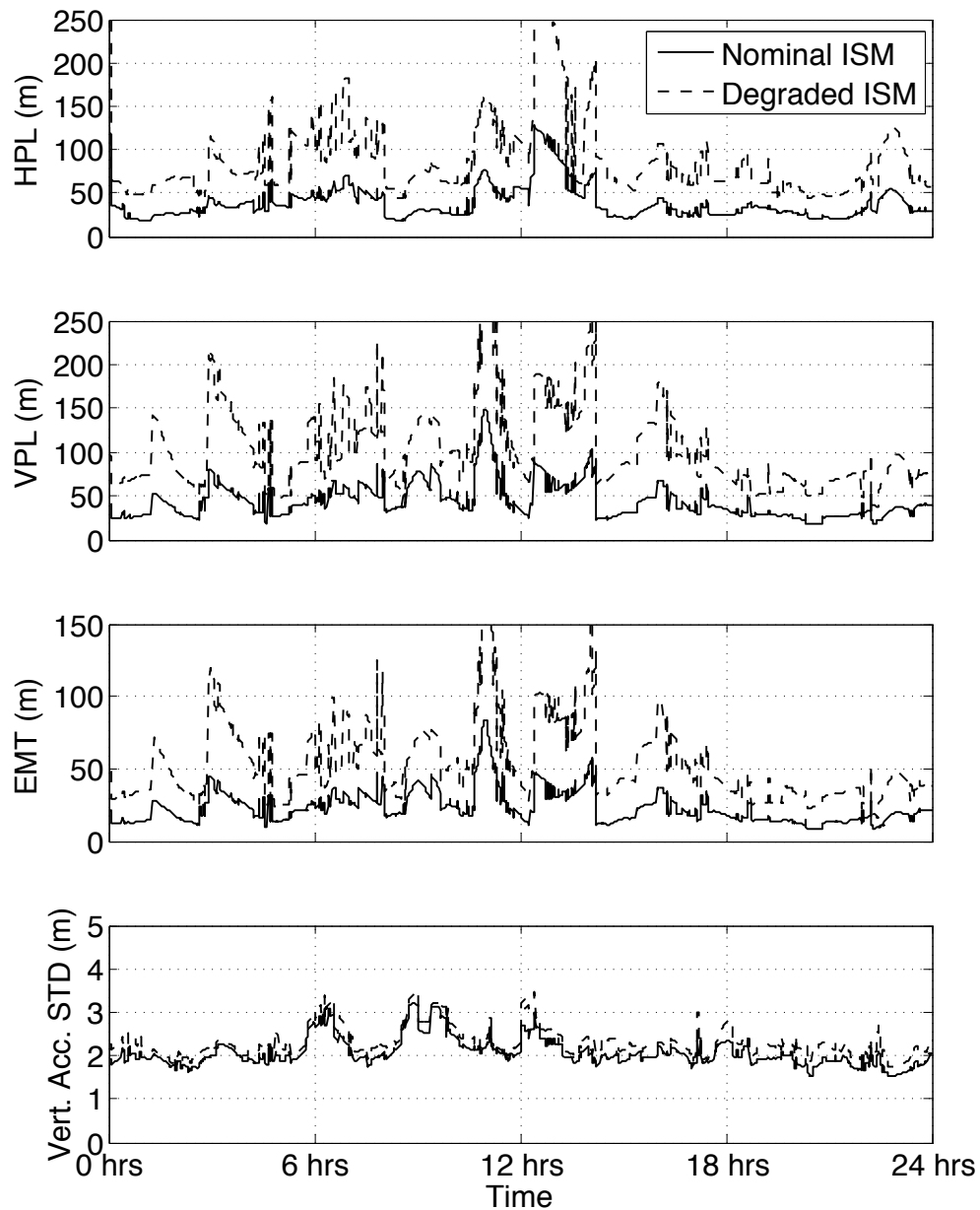


Figure 5.5. Effect of Degrading GLONASS Performance and Reliability on V-ARAIM Performance.

Again, the GPS constellation fault mode generally produces the largest detection thresholds, and because GLONASS URAs are inflated, the position estimate for this mode (which uses only GLONASS and Galileo measurements) has an even larger covariance. Furthermore, the inflated nominal biases have an even greater impact

on performance here than did the biases from the nominal ISM because they are so large for GLONASS. Finally, increasing the prior probability of GLONASS satellite fault increases protection levels because the integrity risk from each mode monitoring GLONASS satellite faults is conditioned on the prior probability of occurrence, which is now an order of magnitude larger.

It is important to remember here that these results are not meant to indicate the current capabilities of ARAIM; they only demonstrate the effects of degrading the SIS performance and reliability on availability performance. As mentioned previously, it is crucial to use ISM parameters that are not so conservative as to degrade availability, but not optimistic so as to compromise the safety of the navigation system. Future GNSS performance is anticipated to be much better than it was conservatively regarded here, and ARAIM performance will reflect this.

CHAPTER 6

IMPROVEMENTS

In real-time applications, every step in the flow chart in Figure 3.1 must be performed at each measurement epoch. This means that the GNSS sampling rate is limited by the time it takes to compute a position estimate, determine associated protection levels, and execute the FDE algorithm. In its entirety, the ARAIM algorithm can potentially become very computationally expensive because the number of modes that the user is required to monitor may increase dramatically as the number of satellites in view increases. This increase is in the number of modes for V-ARAIM is shown in Figure 6.1 using the probabilities of satellite and constellation fault for V-ARAIM given in Table 4.2. In this section, improvements to the V-ARAIM algorithm are investigated, as V-ARAIM is of particular interest in this work. Recall that the user must monitor a sufficient number of modes such that the probability of occurrence of an unmonitored mode is less than the unmonitored probability requirement P_{thres} , which, for V-ARAIM, is 8×10^{-8} . Each value on the x -axis considers all possible combinations of satellites from three constellations. The corresponding value on the y -axis is the maximum number of modes that must be monitored for all those possible combinations. For example, if there are ten GPS and ten GLONASS satellites in view (20 total), monitoring multiple-fault modes is not necessary. However, if there are ten GPS, six GLONASS, and four Galileo satellites in view (again, 20 total), monitoring multiple-fault modes is necessary, and thus the corresponding value on the y -axis reflects this.

Because each monitored mode requires computation of a position estimate, detection thresholds, and exclusion thresholds, the most obvious way of reducing computational time is to reduce the number of monitored modes. This work postu-

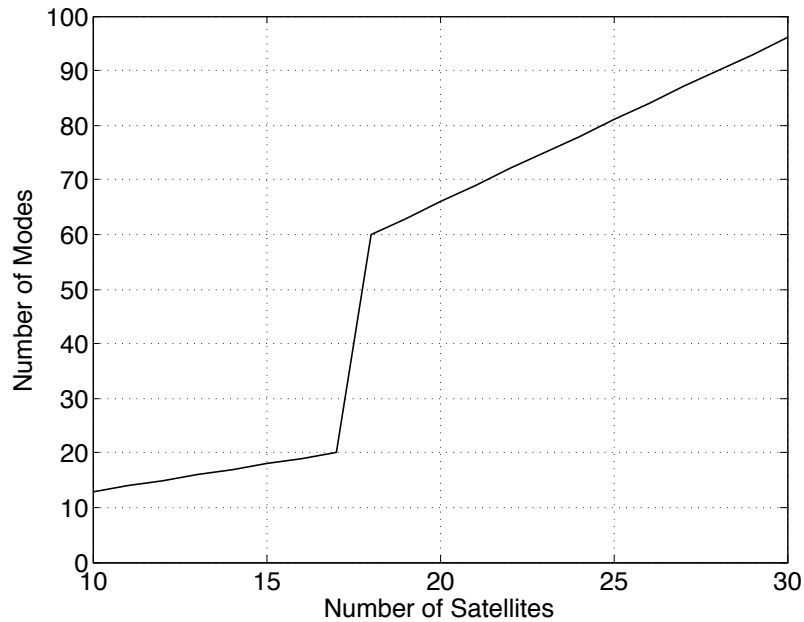


Figure 6.1. Number of Satellites Versus Number of Monitored Modes for V-ARAIM.

lates two methods of doing so and demonstrates the effects on both efficiency and performance. The first method is satellite selection, in which the user selects a subset of satellites to use in estimation, protection level calculation, and FDE out of the all-in-view set. The second method is fault mode grouping, in which multiple monitored modes are grouped together. The efficiency and performance of the ARAIM algorithm using either of these two methods is compared to the efficiency and performance of the ARAIM algorithm with no satellite selection or fault mode grouping, which will henceforth be referred to as the baseline algorithm.

6.1 Satellite Selection

In satellite selection, the user determines the “best” satellites to use in positioning, protection level calculation, and FDE out of the all-in-view subset, as shown in Figure 6.2, subject to a constraint on the number of satellites the user can select. In this example, five satellites are selected out of the eight in view. In practice, the pro-

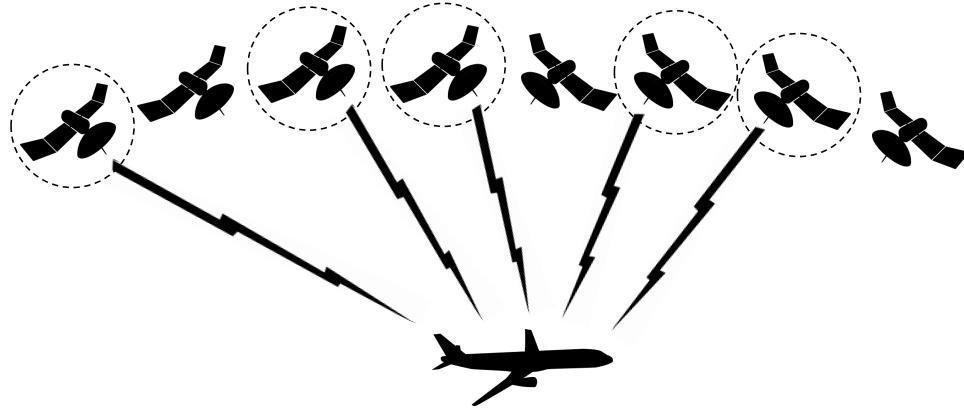


Figure 6.2. Satellite Selection Concept.

cedure of selecting satellites takes place before position estimation; the user receives signals from all of the satellites in view, selects a subset of satellites, and computes a position estimate using this subset. Satellite selection is performed by defining an objective function to minimize. However, there are many possible objective functions to define and many ways of either optimally or sub-optimally selecting satellites that minimize them. Thus, the important aspects of satellite selection are

1. Determining how many satellites must be selected,
2. Deciding from which constellation(s) the satellites should be omitted, and
3. Defining an objective function to minimize and how to minimize it.

These three items will be addressed next, one by one. The overall goal of satellite selection is to select satellites in some educated manner so as to only require the monitoring of single-fault modes when the baseline algorithm is required to monitor multiple-fault modes. At times during which the baseline algorithm does not need to monitor multiple-fault modes, satellite selection is not necessary.

6.1.1 Number of Satellites to Select. The first task of the user is to determine how many satellites must be omitted in order to only require monitoring single-fault

modes. In this work, the values of P_{sat} and P_{const} are fixed (i.e. do not change over time), and the probabilities of satellite fault are the same for all satellites within each individual constellation. Under these simplified conditions, it is a straightforward task to determine the maximum number of allowable satellites. Figure 6.1 shows the maximum number of allowable satellites when using three constellations for V-ARAIM to be 17 using the fault probabilities listed in Table 4.2.

However, in real-time applications with potentially changing ISM parameters, this number may change and thus must be determined by the user at each epoch. Furthermore, in scenarios in which the probabilities of satellite fault vary within an individual constellation, the determination of how many satellites to omit may not be straightforward. For example, the user may decide to keep a satellite with a high fault probability because it is crucial to geometric diversity, in which case a large number of other satellites must be removed. Alternatively, if the user omits this particular satellite, he or she may be required to remove a lesser number of other satellites. In any case, the user must determine how many satellites to omit in general based on the contents of the ISM. For the sake of this analysis, the fault probabilities given in Table 4.2 are used, and thus the selection algorithm will select 17 satellites for V-ARAIM.

6.1.2 Constellations From Which to Remove Satellites. If there are n_{sat} satellites from which the user desires to select k , determination of the optimal subset requires a search through $\binom{n_{sat}}{k}$ possible subsets. For typical satellite numbers using multiple constellations, and in anticipation of more satellites available in the future, this number may be prohibitively large. For example, if the user wishes to select 17 satellites from ten GPS, seven GLONASS, and four Galileo satellites in view, this would require a search through 5,985 possible subsets.

The logical step to take in order to ease the search process is to treat each

constellation individually (i.e. select satellites from each constellation independently of which satellites are selected from other constellations). This inherently introduces sub-optimality into the selection process, but it is motivated by the fact that performance is driven by the GPS constellation fault mode. In general, the constellation fault modes drive performance more than the satellite fault modes, and thus may be treated, to some extent, independently. Now, instead of searching through $\binom{n_{sat}}{k}$ subsets, the user only needs to search through $\binom{n_{GPS}}{k_{GPS}} + \binom{n_{GLO}}{k_{GLO}} + \binom{n_{GAL}}{k_{GAL}}$ subsets, with the constraint that $k_{GPS} + k_{GLO} + k_{GAL} = k$. For typical satellite numbers, this may be an entirely reasonable number of possible subsets to test. Using the example above, if the user wishes to select eight of the ten GPS satellites, six of the seven GLONASS satellites, and three of the four Galileo satellites, this would only require a search through 56 possible subsets. This, however, begs the question: How many satellites should be selected from each constellation? In this example, the user could also decide to select nine GPS satellites and four GLONASS satellites and keep all four Galileo satellites, for instance. Here, if the user wishes to consider all possible ways to remove satellites such that there are at least four remaining in each constellation (to permit estimation using that constellation only), this would still require a search through 448 possible subsets.

In this work, knowledge of the GPS constellation fault mode driving performance motivated the removal of only GPS satellites in the selection process. Initially, it may seem counterintuitive to remove GPS satellites because they are “better” than GLONASS or Galileo satellites. However, because there are often more GPS satellites in view than either GLONASS or Galileo satellites, coupled with the fact that GPS satellites tend to yield more accurate position estimates, the removal of GPS satellites will often have a smaller impact on protection levels than the removal of satellites from other constellations. Essentially, this is saying that GPS measurements

are good enough that omitting some of them will have little impact on performance. On the other hand, GLONASS and Galileo broadcast URAs, and thus measurement accuracies, are so poor that the user cannot afford to omit any of them, as they are all extremely valuable when computing a position estimate for the GPS constellation fault mode. Figure 6.3 shows the percent degradation in protection levels when omitting only GPS satellites compared to omitting the optimal number of satellites from GPS and GLONASS by minimizing the objective function defined in the next section.

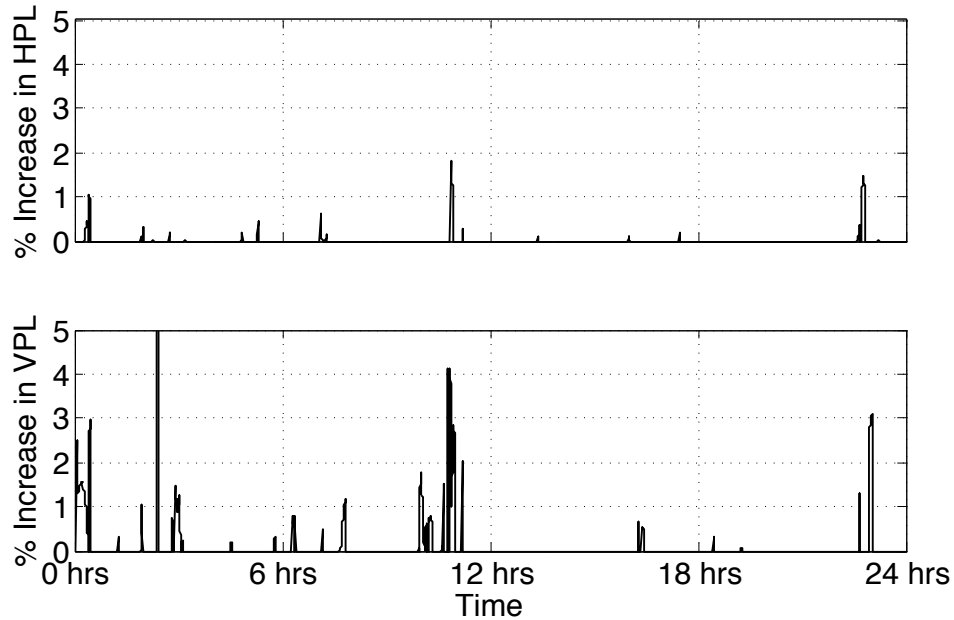


Figure 6.3. Performance Degradation when Omitting GPS Satellites Only Versus Omitting the Optimal Number of GPS and/or GLONASS Satellites.

It can be seen that omitting only GPS satellites causes very little degradation in protection levels, often none at all, and saves the user the task of considering all combinations of satellites to omit from each constellation in order to find the best one. Returning to the example discussed previously, the user would now only be

required to search through 210 possible subsets (there are 210 ways to select six GPS satellites from ten in view). The areas of the plot in which there is no degradation in protection levels are epochs in which omitting only GPS satellites is optimal, which is the vast majority of the time. The results of this heuristic approach indicate that once the user determines how many satellites must be omitted, he or she does not need to explicitly determine from which constellations to remove satellites, as removal of GPS satellites is often optimal and sacrifices little in terms of performance when it is not.

However, as other constellations strengthen and their performances approach the level of GPS, it may be advantageous to remove satellites from these constellations. Furthermore, if the total number of satellites in view becomes so large that the user cannot remove GPS satellites only (because doing so would leave too few remaining), her or she must consider the removal of satellites from other constellations as well.

6.1.3 Objective Function to Minimize. In a strict sense, a subset of satellites can be selected that maximizes performance at each epoch. Even this, however, is ambiguous in that performance is quantified by multiple criteria, such as VPL, HPL, and accuracy. In general, if any performance metric is chosen as the objective function, selecting satellites to minimize it will not necessarily minimize the other performance metrics. In any case, selecting satellites to minimize either the VPL or the HPL is computationally expensive and redundant; if a subset is selected that minimizes VPL, for example, then protection level calculation need not be executed again after position estimation because it already has been executed in the determination of the optimal subset of satellites before estimation. Thus, simpler objective functions are desirable, such as the dilutions of precision (DOPs), pseudo protection levels [20], or a combination of the elements of the all-in-view estimate covariance matrix. A crucial requirement of satellite selection is that the computational time

saved by reducing the number of modes should be greater than the computational time spent selecting satellites; otherwise, overall efficiency would be degraded.

In this work, a particular combination of the estimate covariance matrix entries was deemed a suitable objective to minimize. The objective is a linear combination of a horizontal term and a vertical term, which are defined as

$$C_H^* = 6\sqrt{\frac{1}{2}(C_{1,1} + C_{2,2}) + \sqrt{\frac{1}{4}(C_{1,1} - C_{2,2})^2 + C_{1,2}^2}} \quad (6.1)$$

and

$$C_V^* = 5.33\sqrt{C_{3,3}} \quad (6.2)$$

in [21], where $C_{i,j}$ indicates the i, j element of the covariance matrix

$$C = (G^T W G)^{-1}. \quad (6.3)$$

The objective function is then

$$J = \alpha C_H^* + \beta C_V^*, \quad (6.4)$$

where α and β are coefficients of the horizontal and vertical terms, respectively. While these coefficients are arbitrary, they can be set according to the operational requirements in order to weight the importance of horizontal and vertical geometry. For example, with a HAL of 40 meters and a VAL of 35 meters, α and β may be set as $\frac{35}{40+35}$ and $\frac{40}{40+35}$, respectively (thus, the vertical term is weighted more heavily than the horizontal term because the VAL is more strict than the HAL).

With an objective function defined, there are two ways of minimizing it. The first way is by brute force, in which the objective is evaluated for all $\binom{n_{sat}}{k}$ possible subsets, and the subset with the minimum objective is selected. The second way is by iteration [22], in which the user removes one satellite at a time by minimizing the objective over $\binom{n_{sat}}{n_{sat}-1}$ possible subsets at each iteration until $n_{sat} - 1 = k$.

While the brute force method requires evaluating the objective for $\binom{n_{sat}}{k}$ subsets, the iterative method only requires evaluating the objective for $\sum_{i=k+1}^{n_{sat}} i$ subsets. The iterative procedure trades the optimality of the brute force method for computational efficiency. This tradeoff can be seen in Figure 6.4.

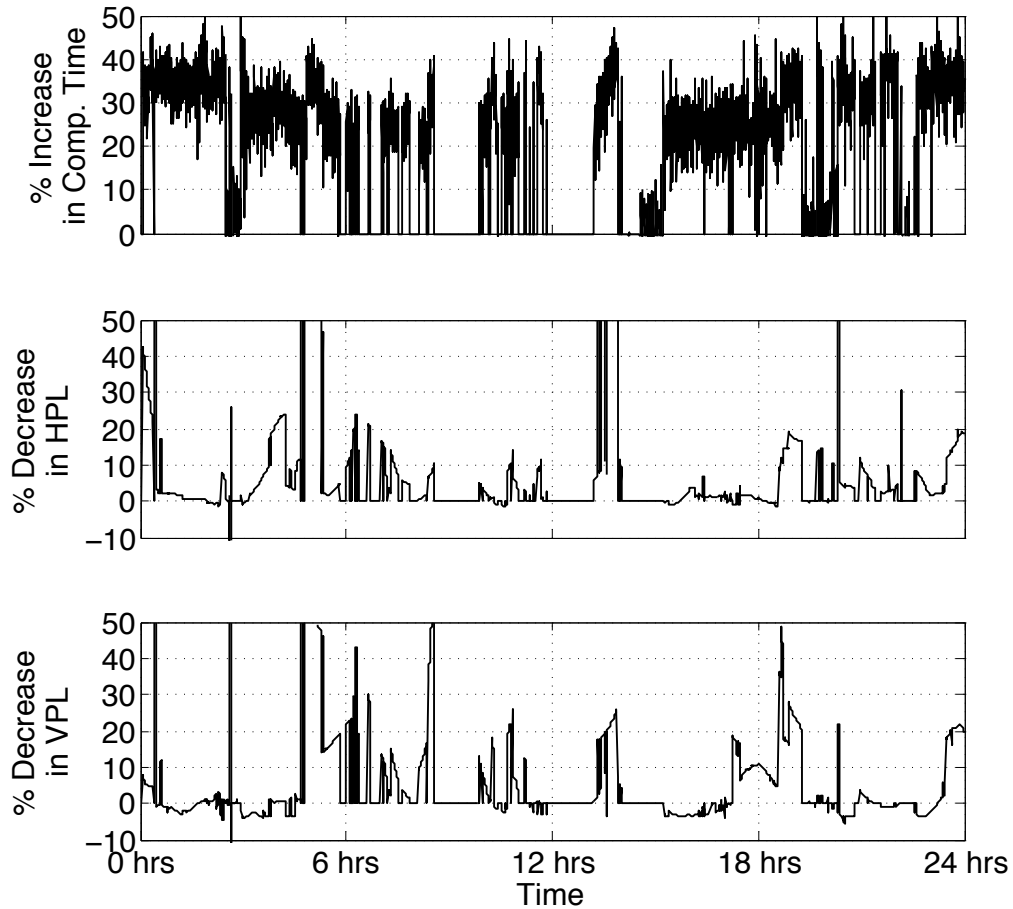


Figure 6.4. Efficiency Degradation and Performance Improvement when Selecting Satellites Optimally Versus Iteratively.

Here, it can be seen that the computational time of the ARAIM algorithm using optimal satellite selection is longer than the computational time of the ARAIM algorithm using iterative satellite selection. However, as we will see later, using opti-

mal selection still dramatically reduces computational time compared to the baseline algorithm. Furthermore, the optimal satellite selection method produces protection levels that are often significantly lower than those produced by the iterative selection method. It is for these reasons that, in this work, the optimal satellite selection method is utilized.

There are two important points to note here. First, remember that the term “optimal” does not refer to a truly optimal selection procedure. The optimality referred to is of the selection procedure of GPS satellites to minimize the given objective function; treating each constellation independently automatically introduces sub-optimality. Second, and crucially, there may be times in which the iterative selection method produces protection levels that are lower than those produced by the optimal selection method. This is because both methods aim at minimizing the given objective, J , not ARAIM protection levels. As such, the iterative selection method can only select satellites that yield an objective equal to or higher than that produced by the optimal selection method, but these satellites may happen to produce lower ARAIM protection levels.

The satellite selection methodology in this work is summarized as follows:

1. Determine the maximum allowable number of satellites, k , so as to only require monitoring single-fault modes.
2. If $n_{sat} \leq k$, satellite selection is not needed. Otherwise, proceed with step three.
3. Generate all $\binom{n_{GPS}}{k - n_{GLO} - n_{GAL}}$ possible subsets of GPS satellites.
4. Evaluate the objective function for each subset, ignoring all other constellations.
5. Select the subset of GPS satellites with the smallest objective. This subset, along with all GLONASS and Galileo satellites in view, is the final satellite

selection.

6. Proceed with position estimation, protection level calculation, and FDE using this final selection.

Figure 6.5 compares the computational efficiency and performance of the ARAIM algorithm using satellite selection with the baseline ARAIM algorithm for V-ARAIM using GPS, GLONASS, and Galileo with the ISM parameters listed in Table 5.3. It also shows the number of (GPS) satellites that the selection algorithm removes in order to only require monitoring single-fault modes. Note the epochs in which no satellites are omitted; at these times, satellite selection is not necessary, and the ARAIM algorithm with satellite selection is the same as the baseline algorithm. The ARAIM algorithm (including estimation, protection level computation, and FDE) using satellite selection reduces computational time by about 60% compared to the baseline ARAIM algorithm due to the reduction in the number of monitored modes. Regarding performance, the satellite selection methodology introduces multiple conflicting performance drivers that manifest themselves in different ways.

The first performance driver is the removal of satellites, which causes positioning accuracy to decrease and tends to increase protection levels. The degradation in vertical accuracy is obvious from the bottom plot in Figure 6.5; by removing satellites, the covariance of the all-in-view position estimate is increased, provided that the SIS performance of the removed satellites is comparable to that of the selected satellites. The correlation between number of satellites removed and degradation in vertical accuracy is relatively strong. Removing satellites also degrades the accuracy of the position estimates for the various modes, increasing σ_k in equation 4.21 and thus causing protection levels to increase. This is most apparent when a large number of satellites are omitted.

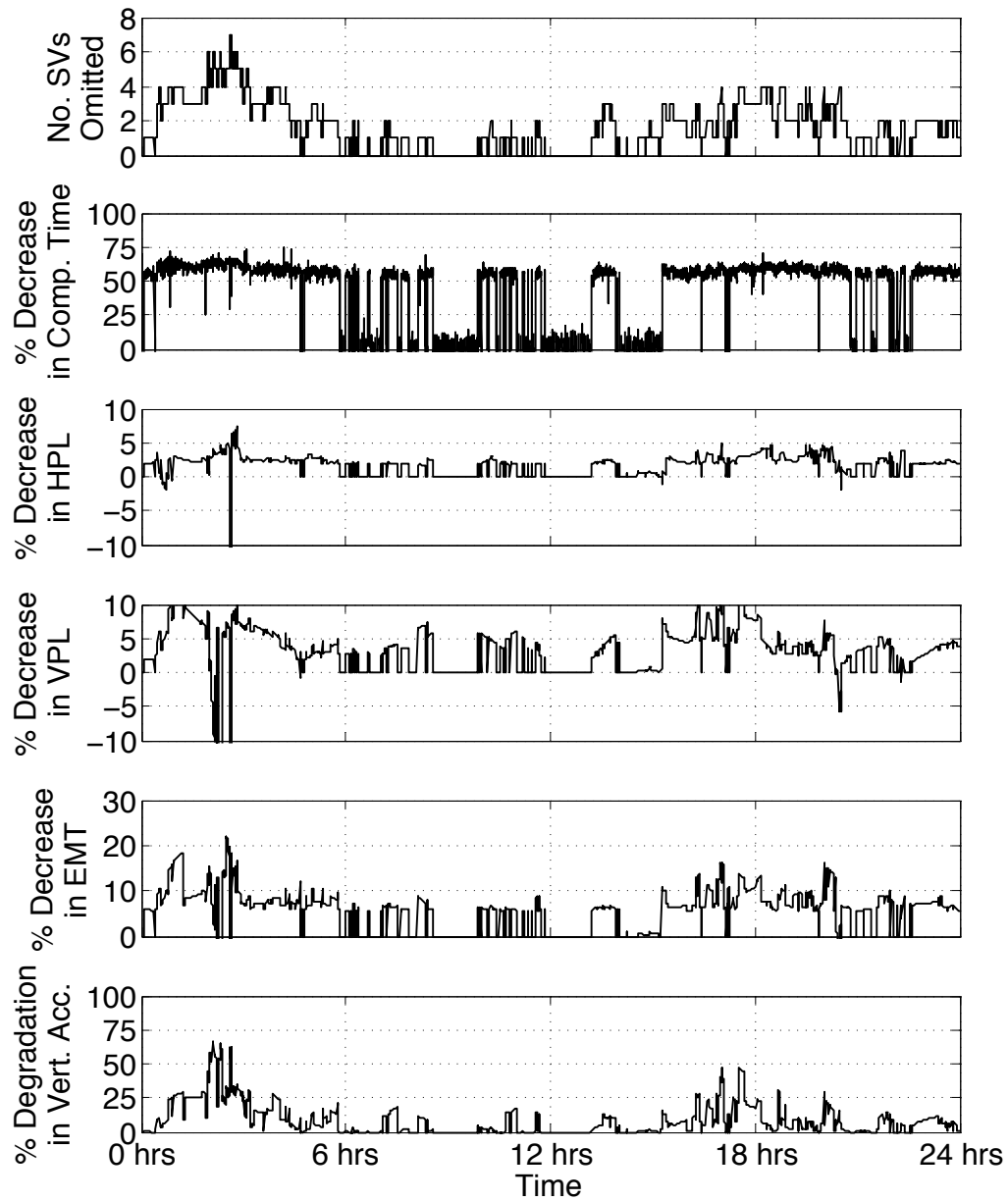


Figure 6.5. Percent Change in Efficiency and Performance of V-ARAIM using Satellite Selection Versus Baseline V-ARAIM using GPS + GLONASS + Galileo.

The second performance driver is the reduction in the number of monitored modes, which acts in conflict to the first performance driver. It contributes to decreasing protection levels and EMTs. By reducing the number of modes, the continuity risk bound used in the determination of the detection thresholds is tightened. This

causes detection thresholds, and thus the EMT, to decrease, as shown in Figure 6.5. The details of this threshold reduction will be discussed in the fault mode grouping section, as its effects are more pronounced and are uninfluenced by any other conflicting performance drivers.

In aggregate, these performance drivers tend to degrade accuracy and cause EMTs and protection levels to improve. At epochs in which a large number of satellites must be removed, protection levels may be degraded, sometimes dramatically so. However, they are consistently improved when removing a modest number of satellites, and as such, it appears that the benefits of reducing the number of monitored modes outweigh the drawbacks of degrading satellite geometry.

Satellite selection provides a way of both dramatically decreasing the computational time of the ARAIM algorithm and marginally improving protection levels. Despite the fact that positioning accuracy is degraded by using satellite selection, anticipated ISM parameters will allow for the use of URAs much lower than those broadcast currently; as such, the degradation in vertical accuracy so apparent in these results may not be so drastic in the future. Furthermore, a significant advantage of satellite selection is that it is flexible; it can be implemented using any number of GNSS, each with varying levels of performance, and it can be tailored to select satellites from any constellation based on any desired performance metric. It should be noted, however, that satellite selection does not alleviate the tracking responsibilities of the receiver. Because satellite selection is performed anew at each epoch, the receiver is still required to track all satellites in view.

6.2 Fault Mode Grouping

The fault mode grouping method reduces the number of monitored modes by grouping them together [23]. While the user may decide to group faults into an

arbitrary number of groups, in this work, modes are grouped according to constellation, meaning that satellite fault modes are grouped together with their respective constellation fault modes. This method of grouping is chosen because it results in the fewest possible number of monitored modes. Grouping in this way is only possible if at least two constellations are being used, since if only one is being used, the constellation fault mode cannot be monitored at all. While satellite selection was employed to prevent the need to monitor multiple-fault modes, fault mode grouping is demonstrated here in a situation in which the baseline algorithm only monitors single-fault modes. Figure 6.6 shows the grouped fault mode matrix corresponding to the example posed in Figure 4.2. In this case, the number of monitored modes is reduced from $n_{sat} + n_{const}$ to n_{const} , as the single-satellite fault modes explicitly monitored in the baseline case are now implicitly monitored in the grouped case.

Mode	GPS					Galileo				
	SV 1	SV 2	SV 3	SV 4	SV 5	SV 1	SV 2	SV 3	SV 4	SV 5
Fault-Free										
GPS Faults										
Galileo Faults										

Figure 6.6. Grouped Fault Mode Matrix.

By grouping satellite fault modes into their respective constellation fault modes, the probability of an unmonitored fault occurring decreases. This is because the grouped fault modes are now monitoring single-constellation faults as well as all com-

binations of satellite faults within that constellation, instead of only single-satellite faults as in the baseline case. Whereas the fault mode probabilities for the baseline case are given by equation 4.7, the fault mode probabilities for the grouped case are given by

$$P_{H_k} = P_{fault,k} \prod_{\substack{j=1 \\ j \neq k}}^{n_{const}} P_{FF,j}, \quad (6.5)$$

where

$$P_{fault,k} = 1 - (1 - P_{const,k}) (1 - P_{sat,k})^{n_{sat,k}}. \quad (6.6)$$

Here, $P_{const,k}$ and $P_{sat,k}$ are the prior probability of constellation k being faulted and the prior probability of any satellite within constellation k being faulted, respectively, and $n_{sat,k}$ is the number of satellites in constellation k . $P_{FF,j}$ is given by

$$P_{FF,j} = (1 - P_{const,j}) (1 - P_{sat,j})^{n_{sat,j}}. \quad (6.7)$$

Again, the assumption is made that the probabilities of satellite fault are the same for all satellites within an individual constellation.

An obvious observation of the fault mode grouping methodology is that it is much simpler to implement than satellite selection procedure, and it requires no steps that introduce sub-optimality like satellite selection does. Figure 6.7 shows the percent improvements in computational efficiency and performance compared to the baseline ARAIM algorithm when using GPS and GLONASS for V-ARAIM using the ISM parameters listed in Table 5.3. Here, performance is only analyzed using dual-constellation ARAIM, as there are currently only two GNSS at FOC. Again, just as in the satellite selection case, the computational time includes estimation, protection level calculation, and FDE. Computational efficiency is improved by about 50% on average due to the reduction in the number of monitored modes. Furthermore, protection levels and EMTs are both consistently reduced.

The reason for this performance improvement, mentioned briefly in the satel-

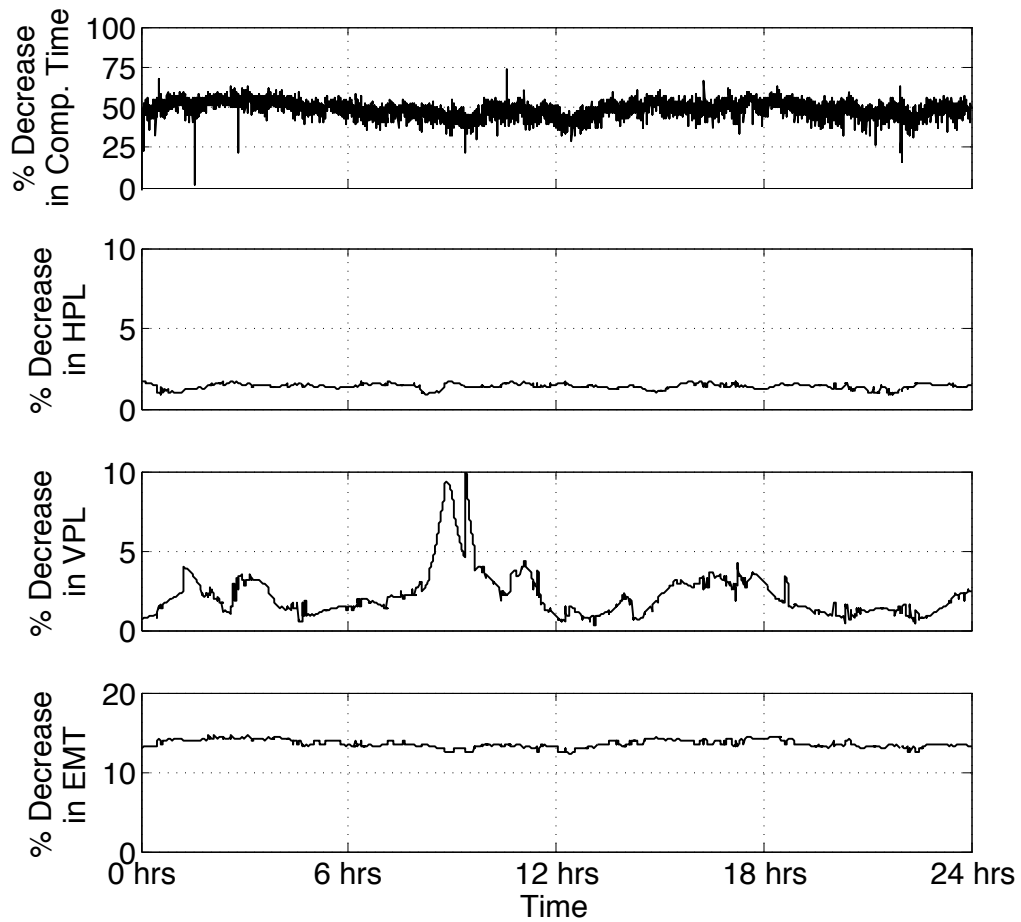


Figure 6.7. Percent Improvements in Efficiency and Performance of V-ARAIM using Fault Mode Grouping Versus Baseline V-ARAIM using GPS + GLONASS.

lite selection section, is as follows. Recall that the continuity risk due to FANE in equation 3.8 is an upper bound due to the over-counting of the medium and dark gray detection regions in Figure 3.4. As the number of modes increases, this bound loosens (becomes more conservative) because more detection regions are over-counted. Conversely, as the number of modes decreases, this bound tightens because there is less over-counting of the detection regions for the various modes. In the trivial case, if there is only one monitored mode, there is no conservatism, and equation 3.8 is the continuity risk due to FANE exactly. In the fault mode grouping case with two constellations, there are only two explicitly monitored fault modes, and so no detec-

tion region is counted more than twice. Again, the bound on the continuity risk due to FANE is still conservative because there are regions in which the continuity risk is double-counted, but recall that the probability of the parity vector lying in these regions under fault-free conditions is small.

This tightening of the bound on the continuity risk due to FANE may be interpreted as a more optimal allocation of the FANE budget between the monitored fault modes, thus causing the detection thresholds to decrease. Essentially, the thresholds decrease because they are not required to compensate for such over-counting of the detection regions in Figure 3.4. All other parameters equal, a decrease in detection thresholds yields a decrease in protection levels. This fact is obvious from observation of equation 4.21. Because the thresholds are decreased, the EMT is also consistently decreased, as shown in the bottom plot in Figure 6.7. Furthermore, there is no degradation of satellite geometry due to satellite removal as there is in the satellite selection procedure. Thus, the performance gain due to this tightening of the continuity risk bound is undiminished by the adverse effects of removing satellites.

Despite these performance improvements, fault mode grouping has limited application. First of all, it only improves performance when constellations with non-negligible P_{const} values are grouped; more specifically, performance is only improved when the grouped constellations have prior probabilities of constellation fault that are larger than the prior probabilities of satellite faults within that constellation. This is because grouping automatically monitors the constellation fault mode, whether or not its prior probability is large or small. As such, fault mode grouping unnecessarily monitors the GPS constellation fault mode for H-ARAIM (recall that baseline H-ARAIM is not required to monitor the GPS constellation fault mode because its probability of occurrence is so small), thus degrading performance. For this reason, fault mode grouping is not suggested for H-ARAIM. Also, it may not be suitable for

GPS + GLONASS V-ARAIM because, presently at least, the probability of GLONASS satellite fault may be larger than 10^{-5} .

Furthermore, fault mode grouping may not yield the same performance improvements when using three or more constellations [24]. Because only two full constellations exist at the time of writing, this work does not examine the effects of fault mode grouping on V-ARAIM performance using GPS, GLONASS, and Galileo. Finally, when using fault mode grouping with two constellations, exclusion cannot be implemented. This is because the removal of either mode in an attempt to exclude a detected fault leaves only one mode remaining, on which second layer detection cannot be performed.

However, these results indicate that employment of fault mode grouping in dual-constellation V-ARAIM has a vast upside. Computational time is greatly reduced, measurement redundancy and satellite geometry remain unchanged, and protection levels are consistently improved. Furthermore, it has been suggested in [25] that exclusion may not be required for dual-constellation V-ARAIM. The significant results of this section provide powerful motivation for the use of fault mode grouping in dual-constellation GPS + Galileo V-ARAIM in the future.

CHAPTER 7

CONCLUSION

With the advent of Galileo and the completion of GLONASS, along with the long-standing GPS, the number of GNSS satellites has increased over the past decade or so. This has opened the door for multi-constellation ARAIM to support more stringent phases of flight, namely aircraft approach requiring vertical guidance. This work demonstrates the multi-constellation approach to position estimation using dual-frequency measurements from GPS, GLONASS, and Galileo satellites. It also discusses the intricacies of GLONASS satellite orbit computation and pseudorange calculation, which are fundamentally different than those for GPS and Galileo. Next, this work comprehensively describes the continuity and integrity risks and their conservatism in the context of MHSS ARAIM.

The real-time implementation of the MHSS ARAIM algorithm is then detailed in full. Two crucial inputs to the algorithm, the measurement error covariances and the ISM, are discussed and quantified. This work then describes the generation of fault hypotheses, the calculation of detection and exclusion thresholds to satisfy the continuity risk requirement, protection level calculation to satisfy the integrity risk requirement, and the FDE procedure.

Next, ARAIM results are presented from a real-time prototype employing GPS, GLONASS, and Galileo measurements. Again, this prototype is not meant to indicate current or future ARAIM capability, as it assumes conservative GNSS performances and reliabilities. However, its results give significant insight into important aspects of ARAIM performance, as well as various performance drivers. This prototype is a significant contribution in that, as these constellations continue to develop,

it can be used to analyze current or future ARAIM performance and validate GNSS performance and reliability assertions. This work demonstrates the effects of adding another constellation (GLONASS) to GPS for H-ARAIM and the effects of degrading GLONASS performance for V-ARAIM using GPS, GLONASS, and Galileo. Most notably, this work demonstrates that V-ARAIM protection levels are driven by the GPS constellation fault mode.

Finally, two methods of improving the computational efficiency of the ARAIM algorithm are investigated: satellite selection and fault mode grouping. The knowledge of the GPS constellation fault mode driving performance for V-ARAIM motivated a crucial step in the satellite selection methodology in which satellites can be removed from the GPS constellation exclusively. The results show that satellite selection can significantly decrease computational time and improve protection level performance for V-ARAIM using three constellations. It is also shown that fault mode grouping by constellation can decrease computational time and consistently improve V-ARAIM performance using two full constellations, thus motivating it for use in future GPS + Galileo V-ARAIM applications.

Short term future work includes incorporating GPS L1/L5 measurements as the L5 signal becomes more widely available. Also, as mentioned previously, the protection levels presented in this work account for the detection function only. As such, future work includes the incorporation of the exclusion function into the protection levels. Long term future work includes developing a suitable airborne GLONASS error model and validating GLONASS and Galileo ISM parameters. As mentioned earlier, both the measurement error characteristics and GNSS performances are of the utmost importance to the accuracy and safety of the ARAIM algorithm.

APPENDIX A
CARRIER SMOOTHING EQUATION DERIVATION

The carrier smoothing equation 2.6 is derived from the block diagram in Figure 2.5 as follows. From the block diagram, we have

$$\bar{\rho} = \frac{\rho - \lambda\phi}{\tau s + 1} + \lambda\phi, \quad (\text{A.1})$$

which can be rearranged as

$$\tau s \bar{\rho} + \bar{\rho} = \rho - \lambda\phi + \tau s \lambda\phi + \lambda\phi. \quad (\text{A.2})$$

Converting from the Laplace domain to the time domain yields

$$\tau \frac{d\bar{\rho}}{dt} + \bar{\rho} = \rho + \tau\lambda \frac{d\phi}{dt}. \quad (\text{A.3})$$

This equation is converted into the discrete time domain by introducing a backward difference for the time derivatives as

$$\frac{dx}{dt} = \frac{x_t - x_{t-1}}{\Delta t}, \quad (\text{A.4})$$

where x represents either $\bar{\rho}$ or ϕ , yielding

$$\tau \left(\frac{\bar{\rho}_t - \bar{\rho}_{t-1}}{\Delta t} \right) + \bar{\rho}_t = \rho_t + \tau\lambda \left(\frac{\phi_t - \phi_{t-1}}{\Delta t} \right). \quad (\text{A.5})$$

This can be simplified as

$$\bar{\rho}_t = \frac{\Delta t}{\tau} \frac{1}{1 + \frac{\Delta t}{\tau}} \rho_t + \frac{1}{1 + \frac{\Delta t}{\tau}} (\bar{\rho}_{t-1} + \lambda(\phi_t - \phi_{t-1})). \quad (\text{A.6})$$

Because $\frac{\Delta t}{\tau} \ll 1$, then

$$\frac{1}{1 + \frac{\Delta t}{\tau}} \approx 1 - \frac{\Delta t}{\tau}. \quad (\text{A.7})$$

Applying this to equation A.6 yields equation 2.6, which is

$$\bar{\rho}_t = \frac{1}{M} \rho_t + \frac{M-1}{M} [\bar{\rho}_{t-1} + (\Phi_t - \Phi_{t-1})], \quad (\text{A.8})$$

where $M = \frac{\tau}{\Delta t}$.

APPENDIX B
GLONASS SATELLITE ORBIT COMPUTATION

GLONASS satellite orbits are computed by propagating a set of initial conditions contained in the ephemeris to the desired time epoch. These initial conditions are the initial x , y , and z components of position, velocity, and acceleration in the PZ-90 ECEF frame at the reference time of ephemeris t_{oe} . The initial accelerations are often referred to as lunisolar accelerations. The reference time of ephemeris lies in the middle of the ephemeris interval. The GLONASS ephemeris is updated every 30 minutes; thus, the user must propagate these initial conditions forward and backward in time by 15 minutes.

The GLONASS ICD [9] specifies a set of six ordinary differential equations (ODEs) that are to be integrated in order to compute the satellite locations at the desired time epoch. They are listed below:

$$f_1 = \frac{dx}{dt} = \dot{x} \quad (\text{B.1})$$

$$f_2 = \frac{dy}{dt} = \dot{y} \quad (\text{B.2})$$

$$f_3 = \frac{dz}{dt} = \dot{z} \quad (\text{B.3})$$

$$f_4 = \frac{d\dot{x}}{dt} = \left(\frac{-\mu}{r^3} x \right) + \left[\frac{3}{2} C \frac{\mu r_E^2}{r^5} x \left(1 - 5 \frac{z^2}{r^2} \right) \right] + \ddot{x}_{LS} + \omega_E^2 x + 2\omega_E \dot{y} \quad (\text{B.4})$$

$$f_5 = \frac{d\dot{y}}{dt} = \left(\frac{-\mu}{r^3} y \right) + \left[\frac{3}{2} C \frac{\mu r_E^2}{r^5} y \left(1 - 5 \frac{z^2}{r^2} \right) \right] + \ddot{y}_{LS} + \omega_E^2 y - 2\omega_E \dot{x} \quad (\text{B.5})$$

$$f_6 = \frac{d\dot{z}}{dt} = \left(\frac{-\mu}{r^3} z \right) + \left[\frac{3}{2} C \frac{\mu r_E^2}{r^5} z \left(3 - 5 \frac{z^2}{r^2} \right) \right] + \ddot{z}_{LS}. \quad (\text{B.6})$$

Here, x , y , and z are the satellite locations, \dot{x} , \dot{y} , and \dot{z} are the satellite velocities, \ddot{x}_{LS} , \ddot{y}_{LS} , and \ddot{z}_{LS} are the lunisolar accelerations, r_E is the equatorial radius of Earth, μ is the gravitational parameter of Earth, C is the second zonal harmonic of geopotential, ω_E is the rotation rate of Earth, and r is defined as

$$r = \sqrt{x^2 + y^2 + z^2}. \quad (\text{B.7})$$

The values of r_E , μ , C , and ω_E can be found in [9].

The ODEs are derived from the orbital force model in [8] and will be indexed by $k = 1, 2, \dots, 6$. For the sake of simplicity, the parameters x , y , z , \dot{x} , \dot{y} , and \dot{z} will be designated by the state vector \vec{X} , such that

$$\vec{X} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T \quad (\text{B.8})$$

and

$$\dot{\vec{X}} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & \ddot{x}_{LS} & \ddot{y}_{LS} & \ddot{z}_{LS} \end{bmatrix}^T. \quad (\text{B.9})$$

Because it is impossible to analytically integrate these ODEs, the GLONASS ICD [9] specifies a fourth-order Runge-Kutta numerical integration procedure. This procedure requires discretizing the ephemeris interval into time steps of length h . The numerical integration procedure forward in time from epoch t_n to t_{n+1} for one satellite is as follows:

1. Evaluate

$$\vec{C}_1 = \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} & c_{1,5} & c_{1,6} \end{bmatrix}^T, \quad (\text{B.10})$$

where $c_{1,k}$ is defined as

$$c_{1,k} = f_k \left(\vec{X}_n \right), \quad (\text{B.11})$$

in which \vec{X}_n is the state at epoch t_n .

2. Evaluate

$$\vec{C}_2 = \begin{bmatrix} c_{2,1} & c_{2,2} & c_{2,3} & c_{2,4} & c_{2,5} & c_{2,6} \end{bmatrix}^T, \quad (\text{B.12})$$

where $c_{2,k}$ is defined as

$$c_{2,k} = f_k \left(\vec{X}_n + \frac{h}{2} \vec{C}_1 \right). \quad (\text{B.13})$$

3. Evaluate

$$\vec{C}_3 = \begin{bmatrix} c_{3,1} & c_{3,2} & c_{3,3} & c_{3,4} & c_{3,5} & c_{3,6} \end{bmatrix}^T, \quad (\text{B.14})$$

where $c_{3,k}$ is defined as

$$c_{3,k} = f_k \left(\vec{X}_n + \frac{h}{2} \vec{C}_2 \right). \quad (\text{B.15})$$

4. Evaluate

$$\vec{C}_4 = \begin{bmatrix} c_{4,1} & c_{4,2} & c_{4,3} & c_{4,4} & c_{4,5} & c_{4,6} \end{bmatrix}^T, \quad (\text{B.16})$$

where $c_{4,k}$ is defined as

$$c_{4,k} = f_k \left(\vec{X}_n + h\vec{C}_3 \right). \quad (\text{B.17})$$

5. Compute the state at epoch t_{n+1} as

$$\vec{X}_{n+1} = \vec{X}_n + \frac{h}{6} \left(\vec{C}_1 + 2\vec{C}_2 + 2\vec{C}_3 + \vec{C}_4 \right). \quad (\text{B.18})$$

The process is similar when propagating the ODEs backwards in time, only the time step h is negative in the above procedure. For the purposes of this work, only the position components of the satellite states are of interest. Also, a step size of $h = 1$ second was used, as step sizes too large may introduce significant errors as the propagation time increases [8]. As confirmation that a step size of one second is appropriate, the difference between the computed satellite positions and those positions directly output by the receiver was on the order of millimeters.

It is important to note that this procedure propagates the initial conditions to regularly spaced times within the ephemeris interval. When doing position estimation, however, the satellite locations must be determined at their times of signal emission. If the state is desired at epoch $t_n + \varepsilon$, corresponding to the time of signal emission, then the state at t_n can be used as the initial conditions and propagated by a time step ε to the time of signal emission according to the procedure above.

APPENDIX C
ARAIM AIRBORNE ERROR MODEL

The airborne error models specified here for GPS and Galileo are similar to those from Appendix A in [4], only adopted for the GPS L1/L2 user as opposed to the L1/L5 user. Recall the error equations 4.2 and 4.3. The values of $\sigma_{UR,i}^2$ and $\sigma_{URE,i}^2$ are found from the navigation message and equation 4.1, respectively. The tropospheric term $\sigma_{tropo,i}^2$ is calculated as

$$\sigma_{tropo,i} = 0.12 \frac{1.001}{\sqrt{0.002001 + \left(\sin \frac{\pi\theta_i}{180}\right)^2}}, \quad (\text{C.1})$$

where θ_i is the elevation of satellite i in degrees and $\sigma_{tropo,i}$ is measured in meters.

The user errors $\sigma_{user,i}^2$ are determined differently for GPS and for Galileo. For GPS, the user error for satellite i is defined as

$$\sigma_{user,i} = \sqrt{\frac{f_{L1}^4 + f_{L2}^4}{(f_{L1}^2 - f_{L2}^2)^2}} \sqrt{\sigma_{MP,i}^2 + \sigma_{noise,i}^2}, \quad (\text{C.2})$$

where

$$\sigma_{MP,i} = 0.13 + 0.53 \exp\left(-\frac{\theta_i}{10}\right) \quad (\text{C.3})$$

and

$$\sigma_{noise,i} = 0.15 + 0.43 \exp\left(-\frac{\theta_i}{6.9}\right). \quad (\text{C.4})$$

Here, $\sigma_{user,i}$, $\sigma_{MP,i}$, and $\sigma_{noise,i}$ are measured in meters. The Galileo user error as a function of satellite elevation may be interpolated from the discrete values listed in Table C.1.

For the purposes of this work, the GPS user error model was also used for GLONASS, as there is not yet an airborne ARAIM error model for GLONASS. It is important to note that, for GLONASS satellites, the user errors depend on the signal frequencies, which vary among satellites due to their FDMA nature.

Table C.1. Galileo User Error.

θ_i (deg)	$\sigma_{user,i}$ (m)
5	0.4529
10	0.3553
15	0.3063
20	0.2638
25	0.2593
30	0.2555
35	0.2504
40	0.2438
45	0.2396
50	0.2359
55	0.2339
60	0.2302
65	0.2295
70	0.2278
75	0.2297
80	0.2310
85	0.2274
90	0.2277

APPENDIX D
PROTECTION LEVEL EQUATION DERIVATION

The protection level equation 4.21 is derived from equation 4.20, which is restated here:

$$I_{req} - P_{NM} = P(|\varepsilon_0| > PL|H_0) P_{H_0} + \sum_{k=1}^n P(|\varepsilon_0| > PL \cap |q_k| < T_k|H_k) P_{H_k} + \sum_{e=1}^{n_e} \left[P(|\varepsilon_e| > PL|H_0) P_{H_e} + P(|\varepsilon_e| > PL|H_0) P_{H_0} + \sum_{\substack{j=1 \\ j \neq e}}^n P(|\varepsilon_e| > PL \cap |q_{e,j}| < T_{e,j}|H_j) P_{H_j} \right]. \quad (D.1)$$

We will consider the integrity risk terms from detection and exclusion functions separately. First, consider the detection terms, which are the first two terms on the right side of the equation. These can be split into four terms as

$$I_{req,detect} = P(\varepsilon_0 < -PL|H_0) P_{H_0} + P(\varepsilon_0 > PL|H_0) P_{H_0} + \sum_{k=1}^n P(\varepsilon_0 < -PL \cap |q_k| < T_k|H_k) P_{H_k} + \sum_{k=1}^n P(\varepsilon_0 > PL \cap |q_k| < T_k|H_k) P_{H_k}. \quad (D.2)$$

The first two terms will be designated A and B , respectively, and each individual component of the sums in the last two terms will be designated C_k and D_k , respectively. For example, C_1 is

$$C_1 = P(\varepsilon_0 < -PL \cap |q_1| < T_1|H_1) P_{H_1}. \quad (D.3)$$

Consider a situation in which a fault contributes a positive bias to the estimate error ε_0 . The following derivation is analogous in the case where a fault contributes a negative bias. Under fault-free conditions H_0 , the estimate error is distributed as shown in Figure D.1, where the regions A and B are shown according to equation D.2. First, D_k is upper bounded using the fact that

$$|\varepsilon_0 - \varepsilon_k| \geq \varepsilon_0 - \varepsilon_k. \quad (D.4)$$

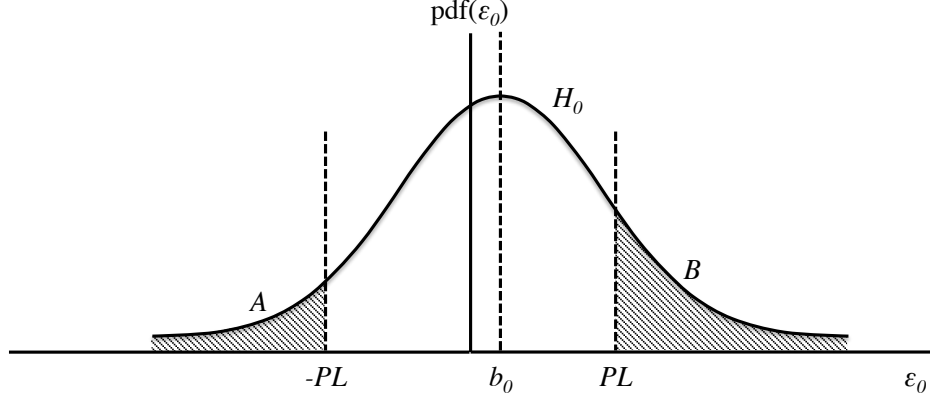


Figure D.1. Estimate Error Distribution under Fault-Free Conditions H_0 .

Note that $q_k = |\varepsilon_0 - \varepsilon_k|$ is the test statistic. Therefore, D_k can be rewritten as

$$D_k = P(\varepsilon_0 > PL \cap |\varepsilon_0 - \varepsilon_k| < T_k | H_k) P_{H_k}, \quad (\text{D.5})$$

which can be upper bounded as

$$D_k \leq P(\varepsilon_0 > PL \cap \varepsilon_0 - \varepsilon_k < T_k | H_k) P_{H_k} \quad (\text{D.6})$$

by using equation D.4. Rearranging yields

$$D_k \leq P(\varepsilon_0 > PL \cap \varepsilon_0 < T_k + \varepsilon_k | H_k) P_{H_k} \quad (\text{D.7})$$

and finally

$$D_k \leq P(\varepsilon_k + T_k > PL | H_k) P_{H_k}. \quad (\text{D.8})$$

Thus, the last term in equation D.2 can be upper bounded as

$$\sum_{k=1}^n P(\varepsilon_0 > PL \cap |q_k| < T_k | H_k) P_{H_k} \leq \sum_{k=1}^n P(\varepsilon_k + T_k > PL | H_k) P_{H_k}, \quad (\text{D.9})$$

and so equation D.2 is upper bounded as

$$I_{req,detect} \leq P(\varepsilon_0 < -PL | H_0) P_{H_0} + P(\varepsilon_0 > PL | H_0) P_{H_0} + \sum_{k=1}^n P(\varepsilon_0 < -PL \cap |q_k| < T_k | H_k) P_{H_k} + \sum_{k=1}^n P(\varepsilon_k + T_k > PL | H_k) P_{H_k}. \quad (\text{D.10})$$

Using an upper bound on the fault-free probability, the second term on the right side of equation D.10 can be upper bounded as

$$P(\varepsilon_0 > PL|H_0) P_{H_0} \leq P(\varepsilon_0 > PL|H_0) \left(P_{H_0} + \sum_{k=1}^n P_{H_k} \right). \quad (\text{D.11})$$

This is a tight bound because $\sum_{k=1}^n P_{H_k}$ is small and P_{H_0} is close to one. The right side of this equation can then be separated into two terms as

$$P(\varepsilon_0 > PL|H_0) \left(P_{H_0} + \sum_{k=1}^n P_{H_k} \right) = P(\varepsilon_0 > PL|H_0) P_{H_0} + P(\varepsilon_0 > PL|H_0) \sum_{k=1}^n P_{H_k}, \quad (\text{D.12})$$

which can be rearranged as

$$P(\varepsilon_0 > PL|H_0) P_{H_0} + P(\varepsilon_0 > PL|H_0) \sum_{k=1}^n P_{H_k} = P(\varepsilon_0 > PL|H_0) P_{H_0} + \sum_{k=1}^n P(\varepsilon_0 > PL|H_0) P_{H_k}. \quad (\text{D.13})$$

Next, by ignoring knowledge of the test statistics, the third term in equation D.10 can be upper bounded as

$$\sum_{k=1}^n P(\varepsilon_0 < -PL \cap |q_k| < T_k|H_k) P_{H_k} \leq \sum_{k=1}^n P(\varepsilon_0 < -PL|H_k) P_{H_k}. \quad (\text{D.14})$$

This may be a loose bound because, if a fault is present, the probability that it is undetected is small. It can be inferred graphically from Figures D.1 and D.2 that, because the fault contributes a positive bias, the right side of equation D.14 is bounded by the second term on the right side of equation D.13. This is because, for all k ,

$$P(\varepsilon_0 < -PL|H_k) \leq P(\varepsilon_0 > PL|H_0), \quad (\text{D.15})$$

where the left side of this equation corresponds to the shaded left tail probability in Figure D.2 for mode k and the right side corresponds to region B in Figure D.1. Thus, using equations D.14 and D.15,

$$\sum_{k=1}^n P(\varepsilon_0 < -PL \cap |q_k| < T_k|H_k) P_{H_k} \leq \sum_{k=1}^n P(\varepsilon_0 > PL|H_0) P_{H_k}, \quad (\text{D.16})$$

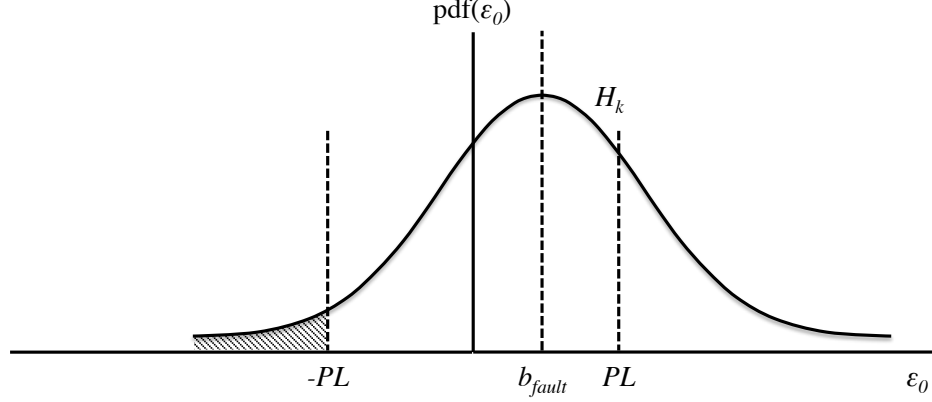


Figure D.2. Estimate Error Distribution under Faulted Conditions H_k .

which can be rearranged as

$$\sum_{k=1}^n P(\varepsilon_0 < -PL \cap |q_k| < T_k | H_k) P_{H_k} \leq P(\varepsilon_0 > PL | H_0) \sum_{k=1}^n P_{H_k}. \quad (\text{D.17})$$

Now equation D.10 has been upper bounded as

$$\begin{aligned} I_{req,detect} \leq & P(\varepsilon_0 < -PL | H_0) P_{H_0} + P(\varepsilon_0 > PL | H_0) P_{H_0} + \\ & P(\varepsilon_0 > PL | H_0) \sum_{k=1}^n P_{H_k} + \sum_{k=1}^n P(\varepsilon_k + T_k > PL | H_k) P_{H_k}. \end{aligned} \quad (\text{D.18})$$

The second and third terms on the right side of equation D.18 can be combined, and the result can be upper bounded as

$$\begin{aligned} I_{req,detect} \leq & P(\varepsilon_0 < -PL | H_0) P_{H_0} + P(\varepsilon_0 > PL | H_0) + \\ & \sum_{k=1}^n P(\varepsilon_k + T_k > PL | H_k) P_{H_k} \end{aligned} \quad (\text{D.19})$$

using the fact that $P_{H_0} + \sum_{k=1}^n P_{H_k}$ is close to one (since P_{NM} is small). Because of the impact of the nominal biases, $B > A$, i.e.

$$P(\varepsilon_0 < -PL | H_0) P_{H_0} \leq P(\varepsilon_0 > PL | H_0) P_{H_0}. \quad (\text{D.20})$$

This is obvious from Figure D.1. Thus, equation D.19 can be bounded by

$$I_{req,detect} \leq P(\varepsilon_0 > PL|H_0) P_{H_0} + P(\varepsilon_0 > PL|H_0) + \sum_{k=1}^n P(\varepsilon_k + T_k > PL|H_k) P_{H_k}. \quad (D.21)$$

This can then be bounded as

$$I_{req,detect} \leq 2P(\varepsilon_0 > PL|H_0) + \sum_{k=1}^n P(\varepsilon_k + T_k > PL|H_k) P_{H_k}. \quad (D.22)$$

by bounding the fault-free probability P_{H_0} with one and combining the first two terms.

This is the upper bound on the integrity risk due to the detection function.

Next, consider the integrity risk due to the exclusion function, which is the third term on the right side of equation D.1 (in brackets). The integrity risk of one exclusion option e is

$$I_{req,e} = P(|\varepsilon_e| > PL|H_0) P_{H_e} + P(|\varepsilon_e| > PL|H_0) P_{H_0} + \sum_{\substack{j=1 \\ j \neq e}}^n P(|\varepsilon_e| > PL \cap |q_{e,j}| < T_{e,j}|H_j) P_{H_j}. \quad (D.23)$$

Combining the first two terms yields

$$I_{req,e} = P(|\varepsilon_e| > PL|H_0) (P_{H_e} + P_{H_0}) + \sum_{\substack{j=1 \\ j \neq e}}^n P(|\varepsilon_e| > PL \cap |q_{e,j}| < T_{e,j}|H_j) P_{H_j}. \quad (D.24)$$

Just like in the detection case, this can be decomposed into four terms as

$$I_{req,e} = P(\varepsilon_e < -PL|H_0) (P_{H_e} + P_{H_0}) + P(\varepsilon_e > PL|H_0) (P_{H_e} + P_{H_0}) + \sum_{\substack{j=1 \\ j \neq e}}^n P(\varepsilon_e < -PL \cap |q_{e,j}| < T_{e,j}|H_j) P_{H_j} + \sum_{\substack{j=1 \\ j \neq e}}^n P(\varepsilon_e > PL \cap |q_{e,j}| < T_{e,j}|H_j) P_{H_j}. \quad (D.25)$$

Using the fact that

$$q_{e,j} = |\varepsilon_e - \varepsilon_{e,j}| \quad (\text{D.26})$$

and

$$|\varepsilon_e - \varepsilon_{e,j}| \geq \varepsilon_e - \varepsilon_{e,j}, \quad (\text{D.27})$$

the fourth term on the right side of equation D.25 can be bounded by

$$\sum_{\substack{j=1 \\ j \neq e}}^n P(\varepsilon_e > PL \cap |q_{e,j}| < T_{e,j} | H_j) P_{H_j} \leq \sum_{\substack{j=1 \\ j \neq e}}^n P(\varepsilon_{e,j} + T_{e,j} > PL | H_j) P_{H_j}. \quad (\text{D.28})$$

This is analogous to that which was done previously in equations D.4 through D.8.

Next, the second term on the right side of equation D.25 can be bounded by

$$P(\varepsilon_e > PL | H_0) (P_{H_e} + P_{H_0}) \leq P(\varepsilon_e > PL | H_0) \left(P_{H_e} + P_{H_0} + \sum_{\substack{j=1 \\ j \neq e}}^n P_{H_j} \right) \quad (\text{D.29})$$

because $j \neq e$ in the sum, and so these events are mutually exclusive. Again, this is a tight bound because $\sum_{\substack{j=1 \\ j \neq e}}^n P_{H_j}$ is small. This can then be split into

$$P(\varepsilon_e > PL | H_0) \left(P_{H_e} + P_{H_0} + \sum_{\substack{j=1 \\ j \neq e}}^n P_{H_j} \right) = P(\varepsilon_e > PL | H_0) (P_{H_e} + P_{H_0}) + P(\varepsilon_e > PL | H_0) \left(\sum_{\substack{j=1 \\ j \neq e}}^n P_{H_j} \right). \quad (\text{D.30})$$

By ignoring knowledge of the second layer test statistics, the third term in equation D.25 can be bounded by

$$\sum_{\substack{j=1 \\ j \neq e}}^n P(\varepsilon_e < -PL \cap |q_{e,j}| < T_{e,j} | H_j) P_{H_j} \leq \sum_{\substack{j=1 \\ j \neq e}}^n P(\varepsilon_e < -PL | H_j) P_{H_j}. \quad (\text{D.31})$$

Just like in the detection case, this may be a loose bound because, if mode j is faulted, the probability of no second layer detection after attempting to exclude mode

e is small. Again, because the fault contributes a positive bias, the right side of equation D.31 can be bounded by the second term on the right side of equation D.30, as

$$\sum_{\substack{j=1 \\ j \neq e}}^n P(\varepsilon_e < -PL|H_j) P_{H_j} \leq \sum_{\substack{j=1 \\ j \neq e}}^n P(\varepsilon_e > PL|H_0) P_{H_j}. \quad (\text{D.32})$$

This bound is analogous to the one in equation D.15. Thus, equation D.25 can be bounded by

$$I_{req,e} \leq P(\varepsilon_e < -PL|H_0) (P_{H_e} + P_{H_0}) + P(\varepsilon_e > PL|H_0) (P_{H_e} + P_{H_0}) + \sum_{\substack{j=1 \\ j \neq e}}^n P(\varepsilon_0 > PL|H_0) P_{H_j} + \sum_{\substack{j=1 \\ j \neq e}}^n P(\varepsilon_{e,j} + T_{e,j} > PL|H_j) P_{H_j}. \quad (\text{D.33})$$

Combining the second and third terms and bounding $P_{H_0} + P_{H_e} + \sum_{\substack{j=1 \\ j \neq e}}^n P_{H_j}$ with one yields

$$I_{req,e} \leq P(\varepsilon_e < -PL|H_0) (P_{H_e} + P_{H_0}) + P(\varepsilon_0 > PL|H_0) + \sum_{\substack{j=1 \\ j \neq e}}^n P(\varepsilon_{e,j} + T_{e,j} > PL|H_j) P_{H_j}. \quad (\text{D.34})$$

This is a tight bound in the same way it was for the detection case. Just as was done for the detection terms, the first term can be bounded as

$$P(\varepsilon_e < -PL|H_0) (P_{H_e} + P_{H_0}) \leq P(\varepsilon_e > PL|H_0) (P_{H_e} + P_{H_0}) \quad (\text{D.35})$$

because $B > A$ (note that these distributions are fault-free - see Figure D.1). Imposing this bound yields

$$I_{req,e} \leq P(\varepsilon_e > PL|H_0) (P_{H_e} + P_{H_0}) + P(\varepsilon_0 > PL|H_0) + \sum_{\substack{j=1 \\ j \neq e}}^n P(\varepsilon_{e,j} + T_{e,j} > PL|H_j) P_{H_j}. \quad (\text{D.36})$$

Combining the first two terms and bounding $P_{H_e} + P_{H_0}$ with one then yields

$$I_{req,e} \leq 2P(\varepsilon_e > PL|H_0) + \sum_{\substack{j=1 \\ j \neq e}}^n P(\varepsilon_{e,j} + T_{e,j} > PL|H_j) P_{H_j}. \quad (D.37)$$

Each individual exclusion option e can be bounded in this way, and thus the total integrity risk due to the exclusion function is bounded by

$$I_{req,exclusion} \leq \sum_{e=1}^{n_e} \left[2P(\varepsilon_e > PL|H_0) + \sum_{\substack{j=1 \\ j \neq e}}^n P(\varepsilon_{e,j} + T_{e,j} > PL|H_j) P_{H_j} \right]. \quad (D.38)$$

In summary, the total integrity risk due to both the detection and exclusion functions is bounded by

$$I_{req} - P_{NM} \leq 2P(\varepsilon_0 > PL|H_0) + \sum_{k=1}^n P(\varepsilon_k + T_k > PL|H_k) P_{H_k} + \sum_{e=1}^{n_e} \left[2P(\varepsilon_e > PL|H_0) + \sum_{\substack{j=1 \\ j \neq e}}^n P(\varepsilon_{e,j} + T_{e,j} > PL|H_j) P_{H_j} \right]. \quad (D.39)$$

This may be represented using Q-functions as

$$I_{req} - P_{NM} \leq 2Q\left(\frac{PL - b_0}{\sigma_0}\right) + \sum_{k=1}^n P_{H_k} Q\left(\frac{PL - b_k - T_k}{\sigma_k}\right) + \sum_{e=1}^{n_e} \left[2Q\left(\frac{PL - b_e}{\sigma_e}\right) + \sum_{\substack{j=1 \\ j \neq e}}^n P_{H_j} Q\left(\frac{PL - b_{e,j} - T_{e,j}}{\sigma_{e,j}}\right) \right], \quad (D.40)$$

which is the protection level equation 4.21.

APPENDIX E
METHOD OF SOLVING PROTECTION LEVEL EQUATION

The protection level equation 4.22,

$$I_{req,d} \left(1 - \frac{P_{NM}}{I_{req}} \right) \leq 2Q \left(\frac{PL_d - b_{0,d}}{\sigma_{0,d}} \right) + \sum_{k=1}^n P_{H_k} Q \left(\frac{PL_d - b_{k,d} - T_{k,d}}{\sigma_{k,d}} \right) + \sum_{e=1}^{n_e} \left[2Q \left(\frac{PL_d - b_{e,d}}{\sigma_{e,d}} \right) + \sum_{\substack{j=1 \\ j \neq e}}^n P_{H_j} Q \left(\frac{PL_d - b_{e,j,d} - T_{e,j,d}}{\sigma_{e,j,d}} \right) \right], \quad (\text{E.1})$$

can be solved in any of the three directions d using the iterative procedure described below.

1. Define the left side of the equation to be

$$P_{HMI,adj,d} = I_{req,d} \left(1 - \frac{P_{NM}}{I_{req}} \right), \quad (\text{E.2})$$

which is the adjusted HMI budget allocated to direction d .

2. Determine the lower bound of the protection level as

$$PL_{min,d} = \max [PL_{min,FF,d}, PL_{min,MD,d}, PL_{min,UE,d}, PL_{min,CE,d}, PL_{min,WE,d}], \quad (\text{E.3})$$

where FF is fault-free, MD is missed detection, UE is unnecessary exclusion, CE is correct exclusion, and WE is wrong exclusion. These terms are calculated from the following equations:

$$PL_{min,FF,d} = Q^{-1} \left(\frac{P_{HMI,adj,d}}{2} \right) \sigma_{0,d} + b_{0,d} \quad (\text{E.4})$$

$$PL_{min,MD,d} = \max_k \left[Q^{-1} \left(\frac{P_{HMI,adj,d}}{P_{H_k}} \right) \sigma_{k,d} + b_{k,d} + T_{k,d} \right] \quad (\text{E.5})$$

$$PL_{min,UE,d} = \max_e \left[Q^{-1} \left(\frac{P_{HMI,adj,d}}{2} \right) \sigma_{e,0,d} + b_{e,0,d} \right] \quad (\text{E.6})$$

$$PL_{min,CE,d} = \max_e \left[Q^{-1} \left(\frac{P_{HMI,adj,d}}{2P_{H_e}} \right) \sigma_{e,0,d} + b_{e,0,d} \right] \quad (\text{E.7})$$

$$PL_{min,WE,d} = \max_{\substack{e,j \\ j \neq e}} \left[Q^{-1} \left(\frac{P_{HMI,adj,d}}{P_{H_j}} \right) \sigma_{e,j,d} + b_{e,j,d} + T_{e,j,d} \right]. \quad (\text{E.8})$$

3. Determine the upper bound of the protection level as

$$PL_{max,d} = \max \left[PL_{max,FF,d}, PL_{max,MD,d}, \right. \\ \left. PL_{max,UE,d}, PL_{max,CE,d}, PL_{max,WE,d} \right]. \quad (\text{E.9})$$

These terms are calculated from the following equations:

$$PL_{max,FF,d} = Q^{-1} \left(\frac{P_{HMI,adj,d}}{2n(n-1)} \right) \sigma_{0,d} + b_{0,d} \quad (\text{E.10})$$

$$PL_{max,MD,d} = \max_k \left[Q^{-1} \left(\frac{P_{HMI,adj,d}}{P_{H_k} n(n-1)} \right) \sigma_{k,d} + b_{k,d} + T_{k,d} \right] \quad (\text{E.11})$$

$$PL_{max,UE,d} = \max_e \left[Q^{-1} \left(\frac{P_{HMI,adj,d}}{2n(n-1)} \right) \sigma_{e,0,d} + b_{e,0,d} \right] \quad (\text{E.12})$$

$$PL_{max,CE,d} = \max_e \left[Q^{-1} \left(\frac{P_{HMI,adj,d}}{2P_{H_e} n(n-1)} \right) \sigma_{e,0,d} + b_{e,0,d} \right] \quad (\text{E.13})$$

$$PL_{max,WE,d} = \max_{\substack{e,j \\ j \neq e}} \left[Q^{-1} \left(\frac{P_{HMI,adj,d}}{P_{H_j} n(n-1)} \right) \sigma_{e,j,d} + b_{e,j,d} + T_{e,j,d} \right]. \quad (\text{E.14})$$

4. Compute the mean of the upper and lower bounds as

$$PL_{half,d} = \frac{1}{2} (PL_{min,d} + PL_{max,d}). \quad (\text{E.15})$$

5. Evaluate the integrity risk corresponding to $PL_{half,d}$ as

$$P_{HMI,half,d} = 2Q \left(\frac{PL_{half,d} - b_{0,d}}{\sigma_{0,d}} \right) + \\ \sum_{k=1}^n P_{H_k} Q \left(\frac{PL_{half,d} - b_{k,d} - T_{k,d}}{\sigma_{k,d}} \right) + \\ \sum_{e=1}^{n_e} \left[2Q \left(\frac{PL_{half,d} - b_{e,d}}{\sigma_{e,d}} \right) + \sum_{\substack{j=1 \\ j \neq e}}^n P_{H_j} Q \left(\frac{PL_{half,d} - b_{e,j,d} - T_{e,j,d}}{\sigma_{e,j,d}} \right) \right]. \quad (\text{E.16})$$

6. If

$$\log_{10} P_{HMI,half,d} \geq \log_{10} P_{HMI,adj,d}, \quad (\text{E.17})$$

then let

$$PL_{min,d} = PL_{half,d}. \quad (\text{E.18})$$

Otherwise, if

$$\log_{10} P_{HMI, half, d} \leq \log_{10} P_{HMI, adj, d}, \quad (\text{E.19})$$

then let

$$PL_{max, d} = PL_{half, d}. \quad (\text{E.20})$$

This step halves the interval at every iteration.

7. Repeat steps four through six until

$$PL_{max, d} - PL_{min, d} \leq PL_{TOL}, \quad (\text{E.21})$$

where $PL_{TOL} = 5 \times 10^{-2}$ meters [4].

BIBLIOGRAPHY

- [1] Misra, Pratap, and Per Enge. *Global Positioning System Signals, Measurements, and Performance*. Revised 2nd ed. Ganga-Jamuna Press. Lincoln, MA. 2012.
- [2] Hofmann-Wellenhof, Bernhard, Herbert Lichtenegger, and Elmar Wasle. *GNSS Global Navigation Satellite Systems*. Springer WeinNewYork. 2008.
- [3] *European GNSS (Galileo) Open Service Signal in Space Interface Control Document*. Issue 2.1. November 2015.
- [4] Blanch, Juan, Todd Walter, Per Enge, Young Lee, Boris Pervan, Markus Rippl, and Alex Spletter. "Advanced RAIM User Algorithm Description: Integrity Support Message Processing, Fault Detection, Exclusion, and Protection Level Calculation." *Proceedings of ION GNSS 2012*. Nashville, TN. September 2012. pp. 2828-2849.
- [5] EU-U.S. Cooperation on Satellite Navigation Working Group C - ARAIM Technical Subgroup. *Milestone 3 Report*. Final Version. 25 February 2016.
- [6] RTCA Special Committee 159. *Minimum Operational Performance Standards for Global Positioning System / Wide Area Augmentation System Airborne Equipment*. RTCA DO-229D. Washington, D.C. 13 December 2006.
- [7] *Interface Specification IS-GPS-200*. 24 September 2013.
- [8] Roßbach, Udo. *Positioning and Navigation Using the Russian Satellite System GLONASS*. PhD Thesis. University of the German Federal Armed Forces. Munich, Germany. March 2000.
- [9] *Global Navigation Satellite System GLONASS Interface Control Document*. Ver. 5.1. Moscow, Russia. 2008.
- [10] Military Topographic Department of the General Staff of Armed Forces of the Russian Federation. "*Parametry Zemli 1990 (PZ-90.11) Reference Document*". Moscow, Russia. 2014.
- [11] Zhai, Yawei, Mathieu Joerger, and Boris Pervan. "Requirement and Performance of Horizontal ARAIM Exclusion Algorithm." *Proceedings of ION GNSS+ 2016*. Portland, OR. September 2016.
- [12] Joerger, Mathieu, Fang-Cheng Chan, and Boris Pervan. "Solution Separation Versus Residual-Based RAIM." *NAVIGATION, Journal of The Institute of Navigation*. vol. 61, no. 4. Winter 2014. pp. 273-291.
- [13] EU-U.S. Cooperation on Satellite Navigation Working Group C - ARAIM Technical Subgroup. *Milestone 2 Report*. Final Draft. 28 November 2014.
- [14] *Global Positioning System Standard Positioning Service Performance Standard*. 4th ed. September 2008.
- [15] FAA-E-2892d. *U.S. Department of Transportation Federal Aviation Administration System Specification for the Wide Area Augmentation System*. 28 March 2014.

- [16] Díaz, Santiago Perea, Michael Meurer, Markus Rippl, Boubeker Belabbas, Mathieu Joerger, and Boris Pervan. “URA/SISA Analysis for GPS-Galileo ARAIM Integrity Support Message.” *Proceedings of ION GNSS+ 2015*. Tampa, FL. September 2015. pp. 735-745.
- [17] Walter, Todd, Juan Blanch, Myung Jun Choi, Tyler Reid, and Per Enge. “Incorporating GLONASS into Aviation RAIM Receivers.” *Proceedings of ION ITM 2013*. San Diego, CA. January 2013. pp. 239-249.
- [18] Heng, Liang. *Safe Satellite Navigation with Multiple Constellations: Global Monitoring of GPS and GLONASS Signal-in-Space Anomalies*. PhD Thesis. Stanford University. Stanford, CA. 2012.
- [19] Gunning, Kazuma, Todd Walter, and Per Enge. “Characterization of GLONASS Broadcast Clock and Ephemeris: Nominal Performance and Fault Trends for ARAIM.” *Proceedings of ION PTTI 2017*. Monterey, CA. January 2017.
- [20] Martini, Ilaria, Daniel Gerbeth, Markus Rippl, Michael Felux, and ARAIM Working Group. “Satellite Selection Methodology for Horizontal Navigation and Integrity Algorithms.” *Proceedings of ION GNSS+ 2016*. Portland, OR. September 2016.
- [21] Walter, Todd, Juan Blanch, and Victoria Kropp. “Satellite Selection for Multi-Constellation SBAS.” *Proceedings of ION GNSS+ 2016*. Portland, OR. September 2016.
- [22] Liu, Min, Marc-Antoine Fortin and Réne Jr Landry. “A Recursive Quasi-Optimal Fast Satellite Selection Method for GNSS Receivers.” *Proceedings of ION GNSS 2009*. Savannah, GA. September 2009. pp. 2061-2071.
- [23] Orejas, Martin, and Jakub Skalicky. “Clustered ARAIM.” *Proceedings ION ITM 2016*. Monterey, CA. January 2016. pp. 224-230.
- [24] Walter, Todd, Juan Blanch, and Per Enge. “Reduced Subset Analysis for Multi-Constellation ARAIM.” *Proceedings of ION ITM 2014*. San Diego, CA. January 2014. pp. 89-98.
- [25] Zhai, Yawei, Mathieu Joerger, and Boris Pervan. “Continuity and Availability in Dual-Frequency Multi-Constellation ARAIM.” *Proceedings of ION GNSS+ 2015*. Tampa, FL. September 2015. pp. 664-674.