UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

A NEW ADAPTIVE INTEGRITY MONITOR FOR THE LOCAL AREA AUGMENTATION SYSTEM UTILIZING CLOSED LOOP FEED BACK

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

By

RICK PENDERGRAFT
Norman, Oklahoma
2013

Downloaded from SHAREOK. Persistent Handle URL: https://hdl.handle.net/11244/7908

A NEW ADAPTIVE INTEGRITY MONITOR FOR THE LOCAL AREA AUGMENTATION SYSTEM UTILIZING CLOSED LOOP FEED BACK

A DISSERTATION APPROVED FOR THE
DEPARTMENT OF ENGINEERING

BY

_____
Dr. John Fagan, Chair

_____
Dr. James J. Sluss

_____
Dr. Hillel Kumin

_____
Dr. Chris Ramseyer

_____
Dr. John Dyer

© Copyright by RICK PENDERGRAFT 2013
All Rights Reserved.

The work presented in these pages is dedicated to my children, Isabella and William. If you learn nothing else from your father take this as an example that persistence and perseverance will prevail, so never give up on your dreams.

Acknowledgements

I would like to express my gratitude to all the people who have helped me on my journey in earning this degree. To each of you that nudged or sometime shoved me back on the path when I strayed, thank you.

I would like to express my sincere gratitude to Dr. John Fagan, my committee chair, for guiding me through the turbulent waters of academia. As my mentor, role-model, and friend he has taught me more than I could ever give him credit for, so I will just say thank you.

I would also like to thank my friends in the lab that contributed many hours of their time to help me build and maintain the system while I performed my research. Especially Chad Davis, David Sandmann, Jacob Henderson, and John Dyer, thank you for all your help.

A special thanks goes to my family for all the babysitting, meals prepared, praying, words of encouragement, and never ending support. Thanks mom and dad for always being there when I needed someone to lend a hand.

Nobody has been more important to me in the pursuit of this degree than my beloved wife, Christi, thank you for your patient and support. Without your love and encouragement this would not have been possible.

Table of Contents

Acknowledgements iv

List of Tables viii

List of Figures ix

Abstract xi

Chapter 1 Introduction 1

1.1 Chapter Overview 4

Chapter 2 Background 5

2.1 Satellite Navigation Overview 5

2.2 Global Positioning System 6

2.2.1 GPS Theory 6

2.2.2 GPS Implementation 11

2.3 GPS Error Sources 12

2.4 Differential GPS 14

2.5 Local Area Augmentation System (LAAS) 15

2.5.1 LAAS Overview 15

2.5.2 LAAS Signal-In-Space 18

2.5.3 LAAS Error Sources 21

2.5.4 LAAS Integrity 22

2.5.5 Integrity Risk Hypotheses 25

Chapter 3 System Architecture 30

3.1 Open-Loop LAAS Architecture Overview 30

3.1.1 Reference Station 31

3.1.2 Base Station 34
3.1.3 VHF Data Broadcast (VDB) 35
3.1.4 Airborne Unit 37
3.2 Closed-Loop Architecture 39
3.2.1 Local Monitor 40
3.2.2 Local Monitor Position Calculation 41
3.2.3 Adaptive Integrity Monitor 42
Chapter 4 Experimental Results 45
4.1 Experimental Objective 45
4.2 Experimental Configuration 46
4.2.1 Dynamic Sensitivity Test 46
4.2.2 Multi-Orbit Static Sensitivity Test 52
4.2.3 Real-Time Adaptive Integrity Monitor 54
4.3 Data Collection and Processing 55
4.3.1 Data Collection 55
4.3.2 Data Processing 58
4.4 Experimental Results 60
4.4.1 Multi-Orbit Static Sensitivity 60
4.4.2 Real-Time Adaptive Integrity Monitor 63
4.5 Results Summary 65
Chapter 5 Conclusions and Future Research 67
5.1 Recommendations for Future Research 68
5.2 Summary 68

References 69

Appendix A: Conversion from LLA to ECEF 72

Appendix B: MMR ARINC Data Logger 74

Appendix C: MMR ARINC Labels 79

Appendix D: Course Deviation Indicator Wiring Diagram 87

Appendix E: MMR ARINC Log File Reader 88

Appendix F: CEI-715 93

List of Tables

Table 1 LAAS Service Performance Requirements 23

Table 2 GBAS Service Levels for Approach Service 23

Table 3 Vertical Missed Detection Multipliers 27

Table 4 Lateral Missed Detection Multipliers 27

Table 5 VDB Data Message Format 36

Table 6 VDB Application Data Format 36

Table 7 Threshold values 43

Table 8 MMR ARINC Labels 57

Table 9 CDI Voltage Ranges 58

Table 10 Multi-Orbit Availability with static σ_{pr_gnd} 63

Table 11 Test Results Summary 66

List of Figures

Figure 1 Position from one satellite 8

Figure 2 Position from two satellites 9

Figure 3 Position from three satellites 9

Figure 4 Unreasonable GPS Ranging Solution 10

Figure 5 Local Area Augmentation System 18

Figure 6 Horizontal Error GPS vs LAAS 20

Figure 7 Vertical Error GPS vs LAAS 20

Figure 8 OU LAAS Architecture 30

Figure 9 LAAS Reference Station with Multipath-Limiting Antenna 31

Figure 10 Thales Navigation GG12 OEM GPS Receiver 32

Figure 11 Freewave OEM Spread Spectrum Radio (900MHz or 2.4 GHz) 32

Figure 12 OU Reference Station Placement 33

Figure 13 Telerad (EM9009 A) VDB Transmitter 35

Figure 14 LAAS Airborne Unit 37

Figure 15 Rockwell Collins (GNLU-930) Multi-Mode Receiver (MMR) 38

Figure 16 Course Deviation Indicator (CDI) 39

Figure 17 Closed-Loop Architecture 39

Figure 18 LAAS Local Monitor 40

Figure 19 Adaptive Integrity Monitor Algorithm 44

Figure 20 Navigation Unavailable 47

Figure 21 Navigation Available 47

Figure 22 Dynamic Sensitivity Diagram 48

Figure 23 Dynamic Sensitivity Flowchart 49
Figure 24 Dynamic σpr_gnd Sensitivity Analysis 51
Figure 25 Loss of Availability 51
Figure 26 Availability shown as Percent σ_{pr_gnd} 52
Figure 27 Data logging diagram 56
Figure 28 Vertical Error for Multi-Orbit Sensitivity Test with $\sigma_{pr_gnd} = 0.1$ 61
Figure 29 Vertical Error for Multi-Orbit Sensitivity Test with $\sigma_{pr_gnd} = 0.2$ 61
Figure 30 Vertical Error for Multi-Orbit Sensitivity Test with $\sigma_{pr_gnd} = 0.4$ 62
Figure 31 Vertical Error for Multi-Orbit Sensitivity Test with $\sigma_{pr_gnd} = 0.6$ 62
Figure 32 Adaptive σ_{pr_gnd} Instantaneous 64
Figure 33 Adaptive σ_{pr_gnd} with a Smoothing Filter 65
Figure A1 ECEF and Reference Ellipsoid 72
Figure C1 ARINC Label 110 79
Figure C2 ARINC Label 111 80
Figure C3 ARINC Label 120 81
Figure C4 ARINC Label 121 82
Figure C5 ARINC Label 140 83
Figure C6 ARINC Label 150 84
Figure C7 ARINC Label 273 85
Figure C8 ARINC Label 370 86
Figure D1 Course Deviation Indicator Wiring Diagram 87
Figure E1 MMR ARINC Log File Reader GUI 88
Figure F1 Condor Engineering CEI-715 93

Abstract

This dissertation presents a new set of algorithms and architecture for implementation of a real-time adaptive integrity monitor for a Local Area Augmentation System (LAAS) that utilizes navigation system vertical error in a feedback loop to deterministically set the broadcast integrity parameter $\sigma_{pr_gnd.}$

This unique method for deterministically assessing the error of the ground subsystem of LAAS in real-time and adapting the broadcast integrity parameter $\sigma_{pr_gnd,}$ rather than using current probabilistic models of predicted worst case scenarios to generate a static value of σ_{pr_gnd}, provides an increase in system integrity. As a result of the increase in system integrity, an additional benefit of increased system availability is achieved.

The research presented in this dissertation demonstrates that the new methodology implemented by the adaptive integrity monitor can deliver performance improvements in both integrity and availability of the LAAS Signal-In-Space.

Chapter 1 Introduction

To meet required navigation system performance for precision approach and landing using the Global Positioning System (GPS), augmentation to the position solution provided by standard GPS is required. The augmentation of GPS by utilizing ground based reference receivers to calculate errors specific to a given airport, and broadcasting these error corrections to a specified local coverage area (approximately a 20-30 mile radius), is referred to as a Ground Based Augmentation System (GBAS). The United States implementation of GBAS is known as Local Area Augmentation System (LAAS). According to the Federal Aviation Administration (FAA), LAAS is intended to provide accuracy, integrity and continuity adequate to support all phases and categories of precision navigation, including CAT I, CAT II and CAT III precision approach and landing requirements.

Currently LAAS is only capable of supporting CAT I approach and landing which requires a maximum permissible integrity risk of $2x10^{-7}$ in a 150 sec interval with a vertical alert limit of 10m, according to the Minimum Aviation System Performance Standards (MASPS) for LAAS.

While LAAS has demonstrated the ability to provide accuracy less than one meter in both the horizontal and vertical directions, which would be more than sufficient for CAT II and CAT III approaches, LAAS has challenges meeting the rigorous requirements for integrity, which are $1x10^{-9}$in any 15 second interval in the vertical direction, and $1x10^{-9}$ in any 30 sec interval in the horizontal direction, according to the MASPS.

One of the primary reasons that integrity requirements for CAT II and CAT III approaches have remained difficult to achieve is the uncertainty in the error associated with the ground subsystem of LAAS.

In the current LAAS specification this error is quantified by the broadcast parameter σ_{pr_gnd}. The broadcast σ_{pr_gnd} is defined by the FAA as the standard deviation of a normal distribution that bounds the Signal-In-Space contribution to the error in the corrected pseudorange at the GBAS reference point. It further describes σ_{pr_gnd} as accounting for all equipment and environmental effects, including the received signal power, the local interference environment and any transient error in smoothing filter output, relative to steady-state, caused by ionospheric divergence. In summary σ_{pr_gnd} encompasses all the error associated with the ground subsystem of LAAS.

To date a great deal of research has been accomplished on the probabilistic error and risk models to quantify σ_{pr_gnd} [12, 13, 14, 16, 17]. As a result of this research, current LAAS ground subsystems set σ_{pr_gnd} to a static value that has been determined using these probabilistic models. Typically, σ_{pr_gnd} is artificially inflated as a means of increasing the position error bounds and ensuring integrity for the worst-case scenario. This approach assumes the chosen threat model properly describes and bounds the system.

Another short coming of the current LAAS architecture, as it relates to σ_{pr_gnd} determination, is its inability to adapt in real-time to environmental change, because the LAAS architecture is an open-loop architecture.

Current LAAS architecture refers to the Honeywell SmartPath SLS 4000 system. This system is the currently the only certified CAT I system used for operation

under DO-217 with DO-278 software standards and DO-254 hardware standards being applied to the system's certification. The SLS 4000 has been deployed at several airports in the United States. To date these installations have not been open to general aviation use due to failure to meet integrity and continuity requirements set forth by the FAA. It is with this in mind that this research effort was initiated.

To overcome the limitations of modeling all possible worst-case scenarios that a LAAS ground subsystem might encounter, the research described in this dissertation provides a new method for preforming sensitivity analysis of σ_{pr_gnd} as it relates to Navigation System Error and system availability. The research presented also describes a novel approach to addressing the determination of σ_{pr_gnd} by utilizing the results of the sensitivity analysis to construct a real-time closed-loop adaptive filter that can deterministically calculate the error in the LAAS corrections as it relates to LAAS ground subsystem error. This real-time closed-loop adaptive filter can deterministically set σ_{pr_gnd} to compensate for LAAS ground subsystem errors without over-bounding the navigation solution, thus increasing integrity without compromising accuracy or availability.

1.1 Chapter Overview

Chapter 2 – Background:

- Overview of satellite navigation, GPS and Differential GPS
- Overview of LAAS architecture
- Theory and computations of LAAS Integrity parameters

Chapter 3 – System Architecture:

- Overview of current LAAS open-loop architecture
- Description of LAAS Components
- Design of new LAAS real-time closed-loop system
- Design of new Adaptive Integrity Monitor

Chapter 4 – Experimental Results:

- Experimental objective and configuration
- Description of data collection and reduction
- Experimental Results

Chapter 5 –Conclusions and Future Work

- Conclusions based on experimental results
- Recommendations for Future Work

Chapter 2 Background

2.1 Satellite Navigation Overview

In the last ten years, the capabilities of satellite navigation for use in aircraft navigation has been expanded to more demanding phases of flight, in particular vertical guidance down to 200 ft. [1]. This expanded role of satellite navigation relies heavily on the Global Position System (GPS) and various systems that augment standard GPS.

The vertical accuracy of standard GPS position solutions is bounded by approximately 22 meters with 95% confidence [2], which is not considered adequate for some phases of aircraft operation, namely terminal area functions such as takeoff and landing. As a result the need exists to augment standard GPS to provide improved location accuracy during takeoff and landing. The development of differential GPS (DGPS) techniques to augment standard GPS has significantly improved the accuracy of GPS position solutions allowing terminal area navigation. There are multiple implementations of differential GPS. One type of DGPS is a ground-based augmentation system (GBAS) that utilizes multiple ground-based reference receivers at known locations to determine local GPS error and provide the necessary GPS corrections for a service area. This type of Ground Based Augmentation System is also referred to as a Local Area Augmentation System (LAAS) and thus the two terms are used interchangeably. The intent of LAAS is to provide differential range corrections and integrity (i.e. safety) information for a local airport area (approximately a 20-30 mile radius) thus allowing precision approach, departure, and other terminal area operations. Furthermore the expectation of the Federal Aviation Administration (FAA) is that LAAS can provide the quality of service required by the Minimum Aviation

System Performance Standards (MASPS)[3], functioning as a sole-means aircraft navigation system for a local airport area.

One of the major concerns surrounding the implementation of LAAS is the possibility of transmitting a LAAS signal-in-space (SIS) that contains Hazardous Misleading Information (HMI). HMI is defined as any information that results in a navigation system error that exceeds the specified alert limit for the selected LAAS service level without alerting the user in the time-to-alert.

Current methods for mitigating HMI are focused around probabilistic modeling and integrity broadcast parameter inflation that over-bounds the navigation system based on proposed worst-case-scenarios [5, 12]. This approach to integrity monitoring has the side effect of reducing system availability. The research in this paper describes a real-time analysis and implementation of an adaptive integrity algorithm based on the LAAS system conceptualized in [4]. The research determines the feasibility of a new and novel way to control system integrity, based on real-time evaluation of the LAAS environment without over-bounding the system and reducing availability.

2.2 Global Positioning System

2.2.1 GPS Theory

The fundamental technique used in GPS positioning is that of satellite ranging. Satellite ranging relies on calculating the distance from a user receiver to multiple satellites whose positions are known. Satellite ranging requires a minimum of three satellites to calculate the position of the user receiver. The distance to each satellite is calculated by receiving unique timing codes from each of the three satellites and calculating the time

it took for each timing code to travel from the satellite to the receiver antenna. These timing codes travel at approximately the speed of light (3×10^8 m/s); therefore if the travel time, T_t (s), is known the distance can be calculated from the following equation. These calculated distances are referred to as pseudoranges.

$$\text{Distance (m)} = T_t * 3 \times 10^8 \text{ m/s} \qquad \text{Eq. 1}$$

Because the travel time is a calculated value, it is susceptible to several different types of errors. For this reason it is often necessary to correct for these errors when the user requires a precise position measurement.

With one calculated distance the receiver knows that it is located somewhere on the surface of an imaginary sphere that is centered on the satellite whose distance is known, as depicted in Figure 1.

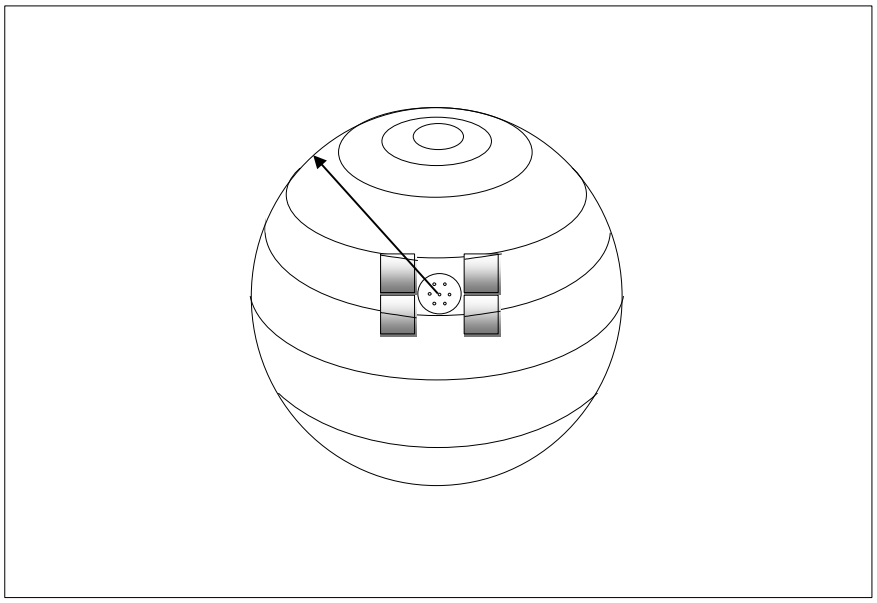

Figure 1 Position from one satellite

A second distance measurement to a different satellite refines the unknown position. This second measurement generates a second sphere that intersects the first. The intersection of two spheres is a circle; therefore the receiver must lie on a circle as shown in Figure 2. A distance measurement from a third satellite narrows the position to two points on a circle as depicted in Figure 3.

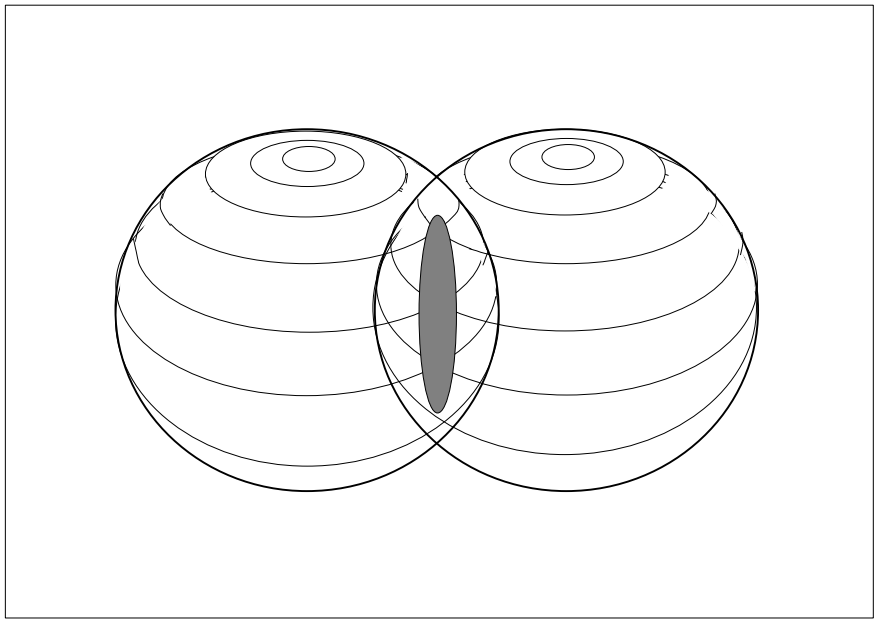

Figure 2 Position from two satellites

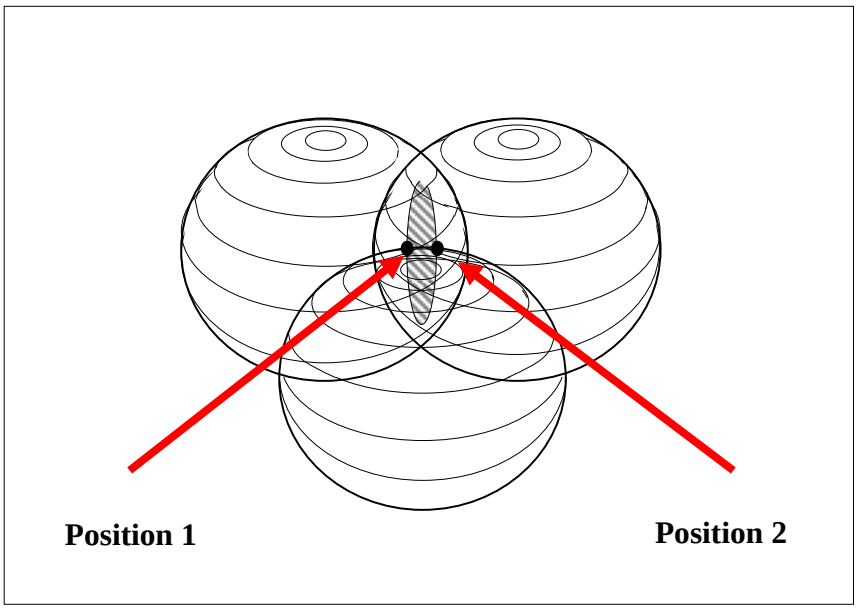

Figure 3 Position from three satellites

A fourth measurement can uniquely identify the location of the unknown position. However, if the desired position is constrained to near the earth's surface then one of the two calculated positions can be discarded because it will furnish the receiver with a position that is not a reasonable solution (i.e. thousands of kilometers away from the earth), while the other position will be near the earth's surface [20]. This is illustrated in Figure 4 below.

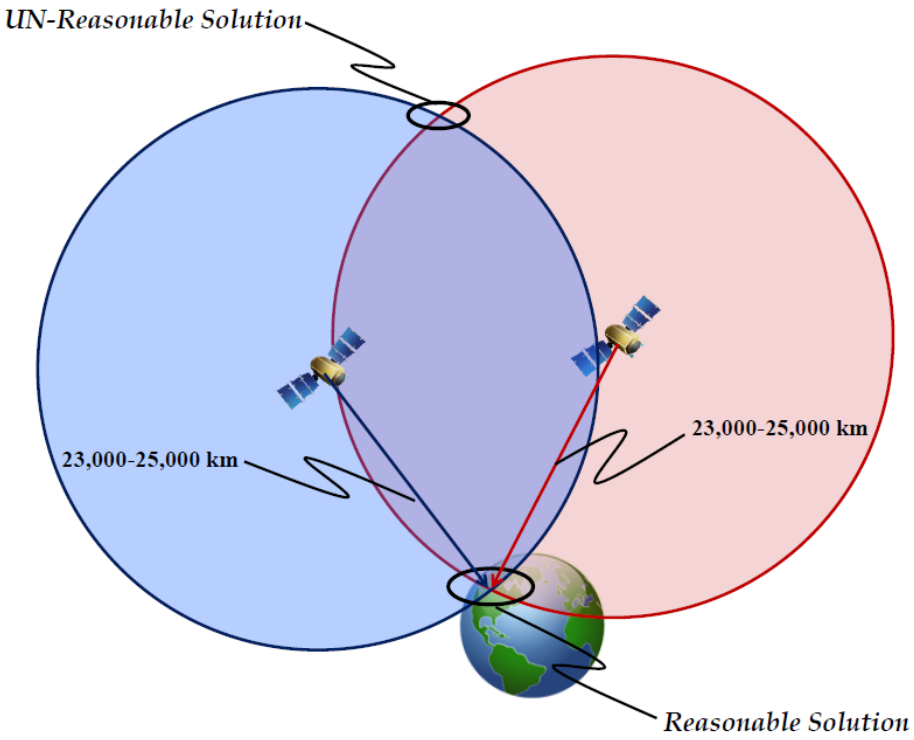

Figure 4 Unreasonable GPS Ranging Solution

2.2.2 GPS Implementation

The Global Positioning System (GPS) is a space based navigation system consisting of three major segments: Space Segment, Control Segment, and User Segment. The Space Segment consists of a constellation of satellites referred to as the Global Navigation Satellite System (GNSS) [5]. The Control Segment monitors the health, timing, and ephemeris of each satellite and uploads corrections and adjustments at regular intervals. The User Segment (or Airborne Segment) consists of a user receiver, which can receive and decode the satellite transmissions.

The GNSS constellation is comprised of six orbital planes, each containing four or more satellites equally spaced within the plane. Each orbital plane is at an approximate inclination angle of 50-60° to the equator. This spacing ensures that a minimum of four satellites can be viewed from any point on the earth's surface at any given time. Each satellite transmits a pseudorandom code (PRN code) that can be used for position measurements. The PRN code incorporates a unique identifier, a time-stamp, and orbital parameters, known as ephemeris [6].

A GPS receiver typically has multiple channels allowing it to track several different in-view satellites that can be used for navigation. GPS navigation is based on satellite ranging: position is determined by measuring the distance from a receiver to all in-view of satellites based on the duration of each transmission to the user.

Each of these distance measurements are referred to as a pseudorange because they are not directly measured but instead are calculated. The receiver obtains the range

to a satellite vehicle (SV) by measuring the difference between satellite PRN code transmit time and the user receive time of the corresponding PRN code and multiplying the time difference by the speed of light. Given four pseudoranges from separate SVs the three dimensional position of the user receiver can be calculated. The need for four pseudoranges arises because there are 4 unknown parameters associated with the position calculation: latitude, longitude, altitude and time. The unknown time parameter is the difference between the GPS clock and the user receiver clock. This time difference, or clock bias, can be resolved by adding a fourth satellite and solving the four equations and four unknowns.

The specified performance of the GPS Standard Positioning Service as it relates to the average position domain accuracy using only the signal in space without any augmentation is a follows:

- Horizontal Error: $\leq$ 9 meters 95% (All-in-View satellites)
- Vertical Error: $\leq$ 15 meters 95% (All-in-View satellites)

This specification does not include errors due to the atmosphere, multipath, or user equipment and is based on a global average [18].

2.3 GPS Error Sources

There are several sources of error associated with GPS navigation. These error sources can generally be categorized as follows: satellite and receiver clock inaccuracies, ephemeris error, and signal propagation delays.

Since the pseudorange calculations are based on transmit and receive time any discrepancy in the satellite and receiver clocks will introduce error into the position

solution. Typically, most clock error is a result of the receiver clock not being as accurate as the atomic clock that is used in the satellite. This difference in clock accuracy is considered constant and can usually be accounted for in a position solution model.

Because satellites drift slightly from their predicted orbits, their exact position relative to the earth is not always known. This uncertainty in position can lead to errors in the receiver's position calculations and is referred to as ephemeris error. Ephemeris error is typically represented as a relatively constant bias that can be modeled and removed from the position calculations.

GPS signal propagation delays are a more variable source of error. One source of signal propagation delay is the atmosphere, specifically the ionosphere and troposphere.

The ionosphere, which is the atmospheric layer located from approximately 80 to 500 kilometers above the earth's surface, contains a large number of free electrons. These electrons appear opaque to the radio waves and therefore cause the signals transmitted from the GNSS satellites to deflect from a straight line path introducing a delay in the signal.

The troposphere, the layer of the atmosphere from ground level to approximately 80 kilometers above the earth's surface, is also a source of interference. The troposphere is a non-dispersive medium and its effect on the GNSS signal is an extra delay in the measurement of the signal traveling from the satellite to the receiver [27]. The delay introduced by the troposphere is dependent on the temperature, pressure and humidity. These factors result in a non-dispersive medium with respect to

electromagnetic waves. The result of the non-dispersive medium is an equal delay to the code and carrier, in contrast to the ionosphere.

Another source of propagation delay is referred to as multipath. Multipath is a result of incoming GPS signals deflecting off of terrestrial objects such as mountains, buildings, large antennas and other obstacles near the user receiver. This deflection in the signal adds a delay to the signal, thus negatively impacting the position calculation.

2.4 Differential GPS

Differential GPS is a method by which GPS signals that contain errors may be augmented to improve the quality of the position solution. Differential GPS relies on the use of a GPS reference receiver at a known location in the vicinity of a user receiver that allows common errors to both receivers to be determined and removed from the user receiver's position solution [7]. Because each user receiver may use a different method of calculating position, the corrections provided by DGPS are given in the range domain.

The reference receiver is a stationary receiver whose exact position is known by means of a precise survey. This reference receiver measures the range to each in-view satellite vehicle (SV). The measured ranges include the actual range to each satellite in addition to the errors discussed previously and is referred to as a pseudorange.

A second range is calculated using the known location (surveyed position) of the reference receiver and the ephemeris based satellite position. This computed range is referred to as the expected range. The expected range and the measured range can then be compared to determine the error in measured pseudorange.

If a user receiver and the reference receiver are using the same set of SVs, the satellite clock error and ephemeris error are common and, to a large degree, the signal propagation errors are common. Thus the calculated error can be applied to the measured range of the remote GPS user receiver and improve position accuracy. This correction is referred to as a pseudorange correction.

As the distance between the two receivers increases a decorrelation in the receiver errors occurs, thus limiting the effectiveness of the reference receiver to a service area of approximately 100 nm [7].

2.5 Local Area Augmentation System (LAAS)

2.5.1 LAAS Overview

Local Area Augmentation System (LAAS) is intended to provide radio navigation for aviation Instrument Flight Rules (IFR) precision approaches and landings from approximately 20nm from the runway threshold, through touchdown and rollout. In addition, LAAS is intended to be suitable for precision navigation in the terminal area including curved approaches and departures and for surface navigation on the airport [8].

The basic principle of LAAS is that pseudorange observations made by ground-based receivers are used to develop differential corrections for each satellite. These corrections are provided to the airborne user's receiver via a VHF data broadcast (VDB). The airborne receiver then applies these corrections in order to produce a set of corrected pseudoranges that are then the basis of a position solution. The underlying assumption is that, for relatively short separations between the ground-based reference

receivers and the airborne-based user receivers, the most significant error sources will be common to both receivers and will therefore be eliminated by differential processing.

Although never observed, it has been opined by researchers that it is possible that in the distance separating the airborne receiver and the reference receiver a waveform can occur that is a moving front in the ionosphere that is traveling at the same speed as the airplane and in the same direction as the airplane, and is between the airborne platform and the reference receiver. This front, referred to as an evil waveform, can cause the reference receiver pseudorange corrections to be correct for the ground station but in error for the airborne platform [12, 16, 17]. It is for this reason that a LAAS station should have an off-site navigator, similar to the OU LAAS implementation. This remote navigator is capable of observing the same pierce points as the airborne receiver and can alert the ground station and the airborne platform of the hazard of such a waveform. Since such a waveform has never been observed at any point on the globe at any time, since the implementation of GPS, it is believed that this threat is very remote but can be mitigated by the remote monitor such as found only on the OU LAAS implementation.

Figure 5 illustrates a typical LAAS, which consists of three main components: space segment, airborne segment, and ground segment. The space segment for LAAS consists of the GNSS satellites. The airborne segment consists of a LAAS enabled GPS receiver. The ground segment is typically composed of the following three components:

1.) A minimum of three precisely surveyed ground reference stations that transmit pseudoranges and other relevant GPS information to the central processing facility.

2.) A central processing facility, referred to as the base station, which receives pseudoranges from all reference stations and uses these pseudoranges to compute estimates of the pseudorange corrections for each satellite signal observed by the reference receivers. The central processing facility also monitors the signal integrity and computes parameters for each satellite that the user may use to determine the availability of the signal in space for a desired level of service and a given satellite geometry [9].

3.) A VHF data broadcast transmitter (VDB) that will broadcast the psuedorange corrections, integrity information, and reference path information to the local area in the 108.0-117.975 MHz band.

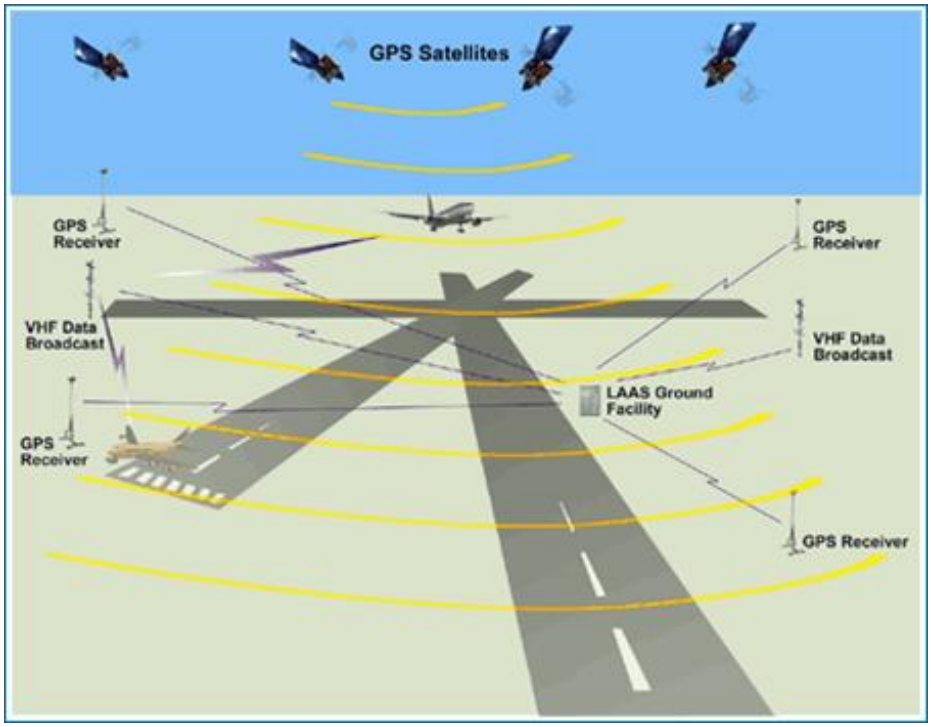

Figure 5 Local Area Augmentation System

2.5.2 LAAS Signal-In-Space

LAAS performance is often characterized by the performance of the "Signal-In-Space" (SIS). The SIS performance is defined in terms of integrity, availability, continuity, and accuracy.

Integrity is a measure of the trust that can be placed in correctness of the information supplied by the system and its ability to provide timely warnings to users when the system should not be used for navigation as a result of errors or failures in the system [3]. Availability is the ability of the navigation system to provide the required function and performance at the initiation of the intended operation (take-off, landing, curved path approach, etc.).

For the LAAS Approach Service, the short-term system availability shall be at least 95%, according to the Minimum Aviation System Performance Standards. Continuity of the system is the ability for the navigation system to provide the required function and performance throughout the entirety of the operation without interruption. Accuracy is the statistical difference, at a 95% probability, between the measured position and known (surveyed) position at any point within the service volume [10, 23]. Current LAAS implementations achieve horizontal accuracy on the order of 1m with 95% confidence and 1.5m with 95% confidence in the vertical direction. Figures 6 and 7 illustrate the achievable accuracy improvement in horizontal and vertical accuracy of LAAS compared to standard GPS navigation. These nominal accuracy numbers easily meet the accuracy requirements in both the vertical and horizontal direction set forth by the International Civil Aviation Organization (ICAO) to support category I (CAT I) operations. However, the challenge for LAAS implementations is meeting the integrity requirements [9]. Current LAAS implementations rely on probabilistic modeling to predict if possible error sources will cause the calculated pseudorange corrections to fail the minimum integrity requirements established by the FAA.

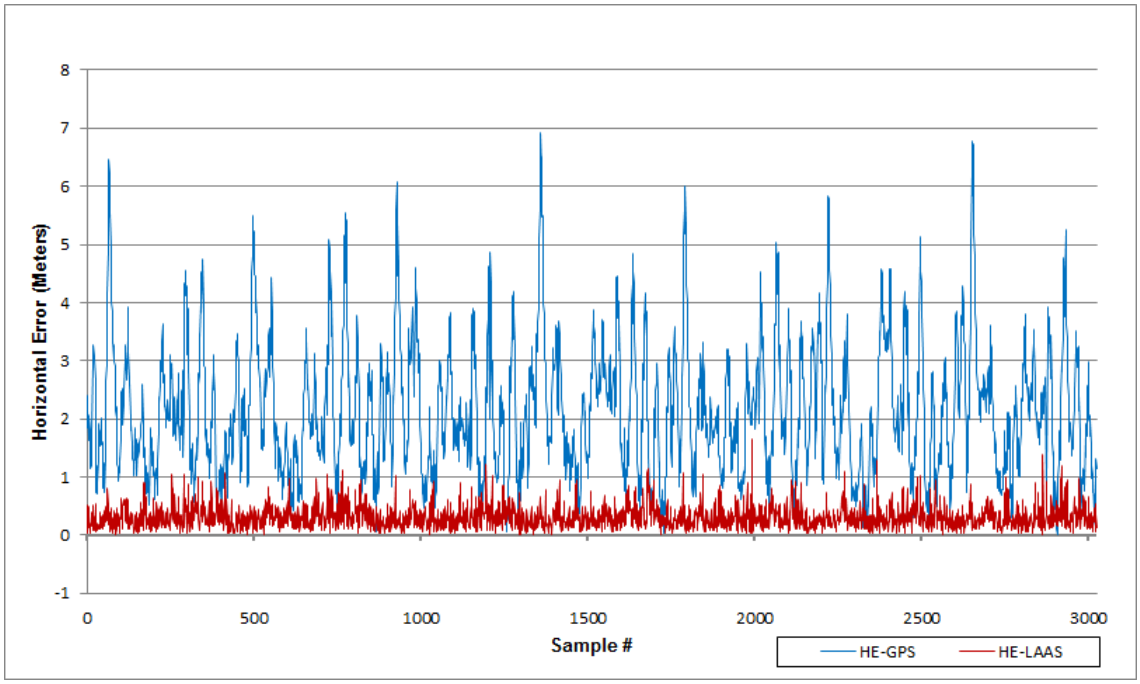

Figure 6 Horizontal Error GPS vs LAAS

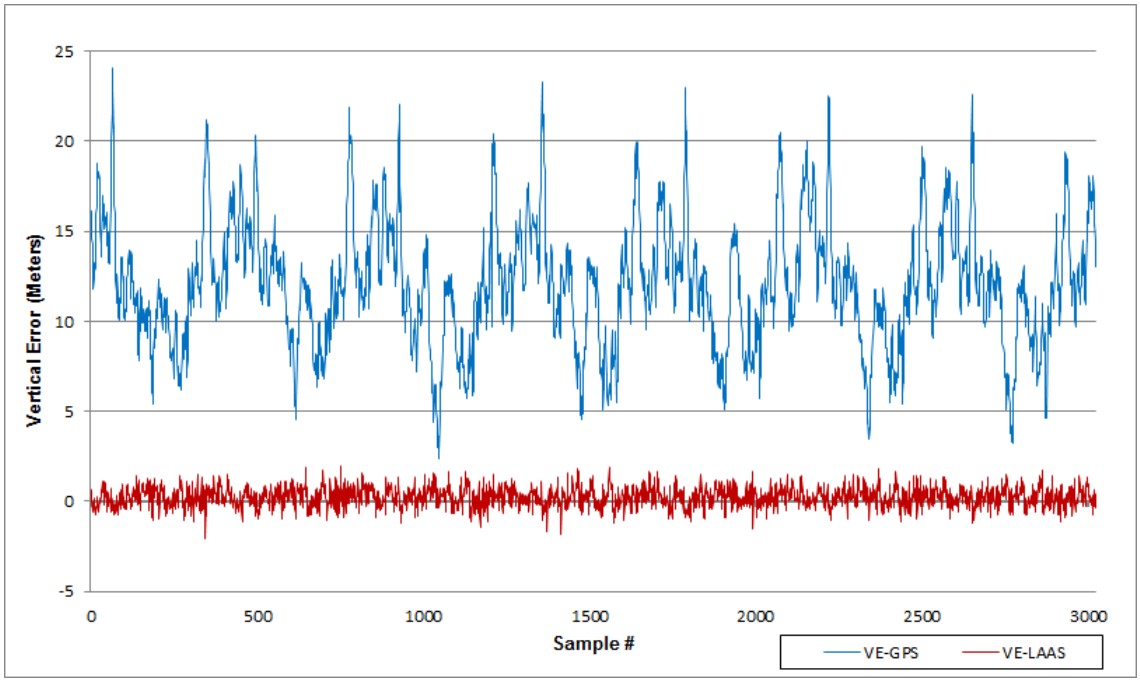

Figure 7 Vertical Error GPS vs LAAS

2.5.3 LAAS Error Sources

LAAS relies on the assumption that error sources encountered by the aircraft receiver and the reference receivers are nearly the same. These errors include ephemeris error, satellite clock error, receiver clock error, and atmospheric error. Ephemeris and clock errors are small contributors to the total error budget, and expected to be zero-mean random processes. Furthermore, satellite clock errors and ephemeris errors are certain to be common to all in-view users [6]. GNSS signal delay caused by the atmosphere, more specifically the ionosphere, is the largest error source for GNSS users, but this error is typically mitigated by applying the LAAS broadcast pseudorange corrections. This mitigation is dependent on the fact that atmospheric delays are highly correlated between the reference receivers and the user receiver. Decorrelation of the error sources between reference receivers and the user receiver are potential sources of Hazardous Misleading Information (HMI) because they are incorporated into the LAAS error correction transmission to the aircraft. The ability to determine these conditions and provide a timely warning to the user, indicating the system should not be used for navigation as a result of the errors, is referred to as integrity monitoring. Integrity information warns of unsafe satellites or conditions and provides a means for aircraft to reliably bound their position errors to the probabilities required for aviation safety [11]. These position bounds are referred to as vertical and horizontal protection limits (VPL and HPL, respectively) and are calculated at the aircraft [15]. Horizontal protection limit is also referred to as Lateral Protection Limit (LPL) and the two terms are used synonymously throughout this document.

2.5.4 LAAS Integrity

In LAAS the quantitative calculation of navigation integrity is accomplished by calculating both vertical and horizontal protection limits at the aircraft. These limits are position bounds that have an associated set of integrity risk parameters based on the categorization of approach within a specific GBAS service level (GSL). The two primary integrity parameters that are used in the quantitative evaluation of LAAS integrity are vertical and lateral alert limits (VAL and LAL, respectively) and are stated in Table 1, following. A description of the typical operations for each of the GBAS service levels referenced in Table 1 can be found in Table 2.

	Accuracy		Integrity				Continuity
GBAS Service Level	Lateral NSE Accuracy 95%	Vertical NSE Accuracy 95%	Integrity Probability	Time to Alert	Lateral Alert Limit	Vertical Alert Limit	Continuity Probability
GSL A	16.0 m (52 ft.)	20.0 m (66 ft.)	$1\text{-}2x10^{-7}$ in any 150 sec	10s	40.0 m (130 ft.)	50 m (160 ft.)	$1\text{-}8x10^{-6}$ in any 15 sec
GSL B	5.0 m (16 ft.)	8.0 m (26 ft.)	$1\text{-}2x10^{-7}$ in any 150 sec	6s	40.0 m (130 ft.)	20 m (66 ft.)	$1\text{-}8x10^{-6}$ in any 15 sec
GSL C	16.0 m (52 ft.)	4.0 m (13 ft.)	$1\text{-}2x10^{-7}$ in any 150 sec	6s	40.0 m (130 ft.)	10 m (33 ft.)	$1\text{-}8x10^{-6}$ in any 15 sec
GSL D	5.0 m (16 ft.)	2.9 m (10 ft.)	$1\text{-}1x10^{-9}$ in any 15 sec vertical, 30 sec lateral	2s	17 m (56 ft.)	10 m (33 ft.)	$1\text{-}8x10^{-6}$ in any 15 sec
GSL E	5.0 m (16 ft.)	2.9 m (10 ft.)	$1\text{-}1x10^{-9}$ in any 15 sec vertical, 30 sec lateral	2s	17 m (56 ft.)	10 m (33 ft.)	$1\text{-}4x10^{-6}$ in any 15 sec
GSL F	5.0 m (16 ft.)	2.9 m (10 ft.)	$1\text{-}1x10^{-9}$ in any 15 sec vertical, 30 sec lateral	2s	17 m (56 ft.)	10 m (33 ft.)	$1\text{-}2x10^{-6}$ in any 15 sec vertical and $1\text{-}2x10^{-6}$ in any 30 sec lateral

Table 1 LAAS Service Performance Requirements

GBAS Service Level	Typical operation(s) which may be supported by the is level of service
A	Approach operations with vertical guidance (performance of APV-I designation)
B	Approach operations with vertical guidance (performance of APV-II designation)
C	Precision Approach to lowest Category I minima
D	Precision Approach to lowest Category IIIb minima, when augmented with other airborne equipment
E	Precision Approach to lowest Category II/IIIa minima
F	Precision Approach to lowest Category IIIb minima

Table 2 GBAS Service Levels for Approach Service

In the LAAS architecture, during approach, the airborne navigation subsystem is required to perform integrity analysis by computing VPL and HPL. The computed protection levels will be compared to the appropriate alert limits for the selected GSL. The airborne sub-system will then indicate if either the HPL or VPL exceeds the alert limit. Typically this is done by full scale deflection of either the horizontal or vertical needle of the course deviation indicator (CDI). Additionally the horizontal and/or vertical flag of the CDI will be displayed indicating the alert limit has been exceeded.

These computed protection levels incorporate integrity data transmitted from the LAAS ground station to the aircraft. These integrity data, referred to as integrity broadcast parameters, represent estimated errors in the system. These estimated errors are represented by B values and Sigma Pseudorange Ground (σ_{pr_gnd}).

The B-values represent pseudorange correction differences across reference receivers. Ideally, the pseudorange corrections from all reference receivers should be the same for a given satellite [19].

Sigma Pseudorange Ground (σ_{pr_gnd}) is defined in [8] as having the purpose of accounting for all equipment and environmental effects, including the received signal power, the local interference environment, and any transient error in smoothing filter output, relative to steady-state, caused by ionospheric divergence for each ranging source (reference receiver).

It can be seen in equations 3, 4, 8 and 9 below that these integrity broadcast parameters impact the computation of VPL and HPL and therefore impact overall integrity of the system.

Due to the wide variety of errors represented by σ_{pr_gnd} it is difficult to quantify this integrity parameter. The guidance provided by [8] for quantifying this parameter is as follows:

The broadcast σ_{pr_gnd} shall be such that the LAAS service availability, as defined in Section 2.3.3.2 of RTCA/DO-245A, for a nominal 24 satellite constellation described in the GPS Standard Positioning Service Performance Standards, must be at least 0.99.

In practice today σ_{pr_gnd} is a static value that is determined using probabilistic models [16,17]. Typically this σ_{pr_gnd} is artificially inflated as a means of increasing the position error bounds and ensuring a known integrity for the worst-case scenario.

2.5.5 Integrity Risk Hypotheses

According to the LAAS Minimum Aviation System Performance Standards the following algorithms for the computation of the integrity protection limits will be implemented by the airborne navigation subsystem.

H0: fault-free condition

H1: single reference receiver failure

These algorithms assume a normally distributed fault-free error model for the broadcast pseudorange corrections. The standard deviation of the correction error is further assumed by the aircraft navigation system to be equal to the broadcast value of σ_{pr_gnd} for each satellite [14].

The protection levels VPL and LPL will be computed as follows, where VPL and LPL are first computed using the H0 hypothesis, and then computed using the H1 hypothesis. The maximum of the two hypotheses will be used for comparison with the alert limits.

$$HPL = MAX\ \{LPL_{H0}, LPL_{H1}\} \quad \text{Eq. 1}$$

$$VPL = MAX\ \{VPL_{H0}, VPL_{H1}\} \quad \text{Eq. 2}$$

VPL and LPL, assuming the H0 hypothesis, shall be computed as follows:

$$VPL_{H0} = K_{ffmd}\sqrt{\sum_{i=1}^{N} S_{vert,i}{}^{2}\,\sigma_i^2} \quad \text{Eq. 3}$$

$$LPL_{H0} = K_{ffmd}\sqrt{\sum_{i=1}^{N} S_{lat,i}{}^{2}\,\sigma_i^2} \quad \text{Eq. 4}$$

where:

K_{ffmd}	multiplier which determines the probability of fault-free missed detection (ffmd) given M reference receivers, possible values are shown in Table 3 and Table 4
$S_{vert,i} = S_{v,i} + S_{x,i} * \tan\theta_{GS}$	projection of the vertical component and translation of the along track errors into the vertical for i^{th} ranging source
$S_{v,i}$	the partial derivative of position error in the vertical direction with respect to pseudorange error on the ith satellite
$S_{x,i}$	the partial derivative of position error in the x-direction with respect to pseudorange error on the ith satellite
$S_{lat,i} = S_{y,i}$	projection of the lateral component for i^{th} ranging source
$S_{y,i}$	the partial derivative of position error in the y-direction with respect to pseudorange error on the ith satellite
θ_{GS}	glide path angle for the final approach path
N	number of ranging sources used in the position solution
i	ranging source index
σ_i	the pseudorange error term for the ith ranging source, as defined in equation 5

GSL	K_{ffmd}			K_{md}		
	M_m=2	M_m=3	M_m=4	M_m=2	M_m=3	M_m=4
A, B, C	5.762	5.810	5.847	2.935	2.898	2.878
D	6.8	6.9	6.9	3.8	3.7	3.7
E	6.8	6.9	6.9	3.8	3.7	3.7
F	6.8	6.9	6.9	3.8	3.7	3.7

Table 3 Vertical Missed Detection Multipliers

GSL	K_{ffmd}			K_{md}		
	M_m=2	M_m=3	M_m=4	M_m=2	M_m=3	M_m=4
A, B, C	5.762	5.810	5.847	2.935	2.898	2.878
D	6.8	6.9	6.9	3.8	3.7	3.7
E	6.8	6.9	6.9	3.8	3.7	3.7
F	6.8	6.9	6.9	3.8	3.7	3.7

Table 4 Lateral Missed Detection Multipliers

$$\sigma_i^2 = \sigma_{pr_gnd,i}^2 + \sigma_{tropo,i}^2 + \sigma_{air,i}^2 + \sigma_{iono,i}^2 \quad \text{Eq. 5}$$

where:

$\sigma_{pr_gnd,i}^2$ is the total (post correction) fault-free noise term provided by the LAAS ground station for satellite *i* (transmitted in Type I message)

$\sigma_{tropo,i}^2$ computed by airborne equipment to cover the residual tropospheric error for satellite *i*

$\sigma_{iono,i}^2$ computed by airborne equipment to cover the residual ionospheric delay uncertainty for the ith ranging source

$\sigma_{air,i}^2$ is the standard deviation of the aircraft contribution to the corrected pseudorange error for the ith ranging source. The total aircraft contribution includes the receiver contribution and a standard allowance for airframe multipath.

VPL and LPL, assuming the H1 hypothesis, shall be computed as follows:

$$LPL_{H1} = MAX\{LPL_i\} \quad \text{Eq. 6}$$

$$VPL_{H1} = MAX\{VPL_i\} \quad \text{Eq. 7}$$

The airborne system will calculate $VPL_{j,H1}$ and $LPL_{j,H1}$ for all j (1 to $MAX\{M_i\}$) as follows:

$$LPL_{j,H1} = \left|\sum_{i=1}^{N} S_{lat,i}\, B_{i,j}\right| + K_{md}\sqrt{\sum_{i=1}^{N} {S_{lat,i}}^2 \sigma_{i,H1}^2} \quad \text{Eq. 8}$$

$$VPL_{j,H1} = \left|\sum_{i=1}^{N} S_{vert,i}\, B_{i,j}\right| + K_{md}\sqrt{\sum_{i=1}^{N} {S_{vert,i}}^2 \sigma_{i,H1}^2} \quad \text{Eq. 9}$$

where:

K_{md}	multiplier which determines the probability of missed detection (md) given that the ground subsystem has faulted 1 reference receivers, possible values are shown in Table 3 and Table 4
$S_{vert,i}$	described above for the H0 hypothesis
$S_{lat,i}$	described above for the H0 hypothesis
j	ground subsystem reference receiver index
$B_{i,j}$	The *B* value for the ith satellite and j^{th} reference receiver. If the jth receiver was not used to produce the ith differential correction, then the ground subsystem will not provide a value for $B_{i,j}$. In this case, the airborne subsystem sets $B_{i,j}$ to zero in the equations below.
M_i	the number of reference receivers used to compute the pseudorange corrections for the ith ranging source (indicated by the *B* values)
$\sigma_{i,H1}$	the pseudorange error term for the ith ranging source, as defined in equation 10.
U_i	the number of reference receivers used to compute the pseudorange corrections for the ith ranging source, excluding the jth reference receiver

$$\sigma_{i,H1}^2 = \frac{M_i * \sigma_{pr_gnd,i}^2}{U_i} + \sigma_{tropo,i}^2 + \sigma_{air,i}^2 + \sigma_{iono,i}^2 \qquad \text{Eq. 10}$$

Chapter 3 System Architecture

3.1 Open-Loop LAAS Architecture Overview

The block diagram in Figure 8 depicts the current architecture of the University of Oklahoma LAAS research facility that will be used to conduct this research. The OU LAAS research facility includes the GNSS constellation, 4 widely spaced ground-based reference stations, LAAS Base Station, VDB Transmitter, and an airborne unit.

This architecture is based on the open loop stochastic approach used by current certified Local Area Augmentation Systems, such as the Honeywell SmartPath SLS 4000, that have been implemented to follow the requirements of FFA specification document (FAA-2937A) [8]. This document establishes the minimum performance requirements for a non-Federal Category I LAAS Ground Facility (LGF).

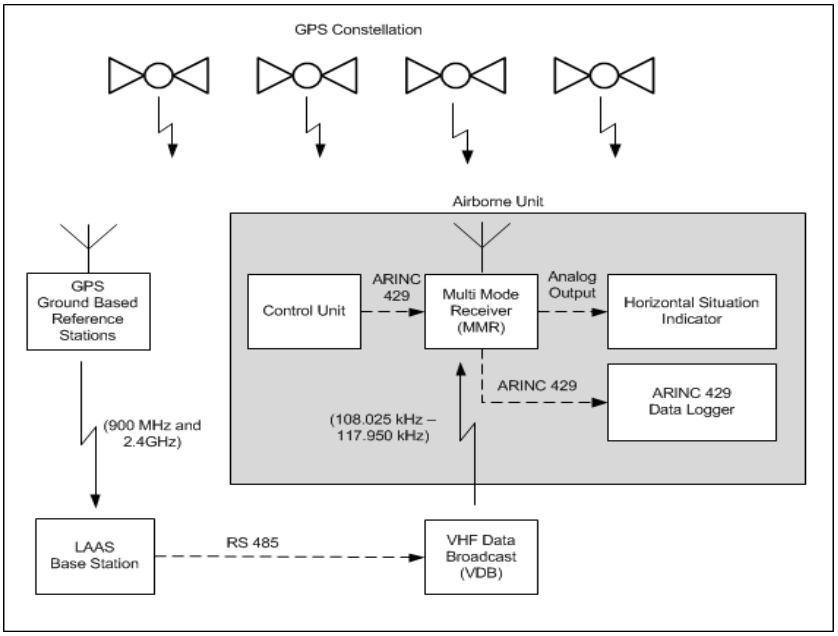

Figure 8 OU LAAS Architecture

3.1.1 Reference Station

Each reference station consists of the following components:

1.) An OU designed multipath-limiting antenna (see Figure 9)

2.) Two GPS receivers (see Figure 10)

3.) Two wireless spread spectrum radios (see Figure 11)



Figure 9 LAAS Reference Station with Multipath-Limiting Antenna

Each reference station is built with redundant GPS receiver and spread spectrum radio pairs, with one spread spectrum radio broadcasting at 900 MHz to the base station and the other broadcasting at 2.4 GHz.

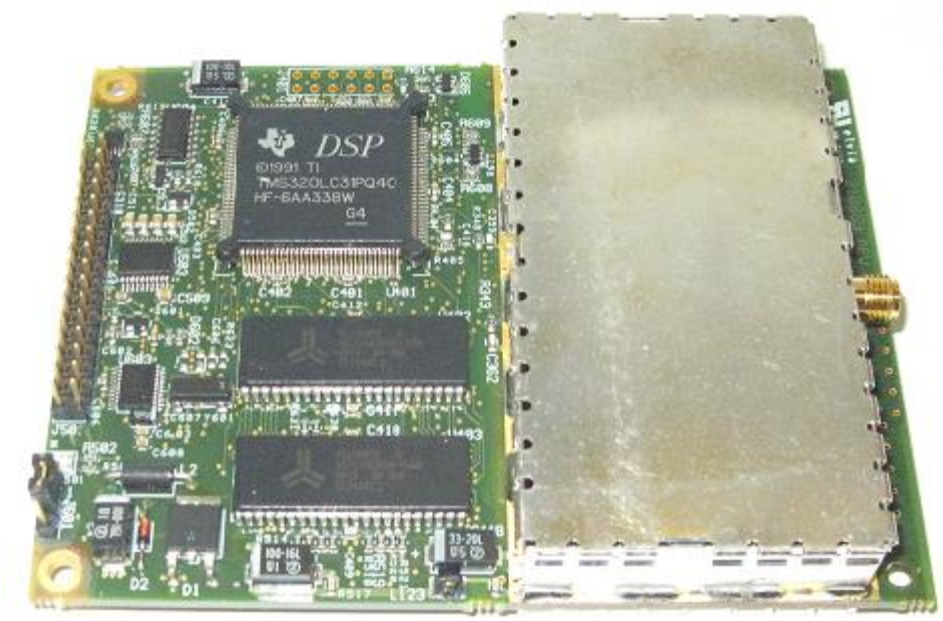

Figure 10 Thales Navigation GG12 OEM GPS Receiver



Figure 11 Freewave OEM Spread Spectrum Radio (900MHz or 2.4 GHz)

The function of the reference station is to collect pseudorange data from the GPS constellation and transmit the data to the LAAS ground station for use in calculating pseudorange corrections. Each reference station receives GPS messages from the GPS constellation. These signals include both ranging signals and navigation

messages (ephemeris, almanac, constellation health, etc.). Each reference station will transmit calculated pseudoranges and the additional navigation messages back to the LAAS ground station via a spread spectrum radio. Figure 12 illustrates the wide placement of the reference stations at the University of Oklahoma LAAS research facility and the local monitor described in section 3.2.1.

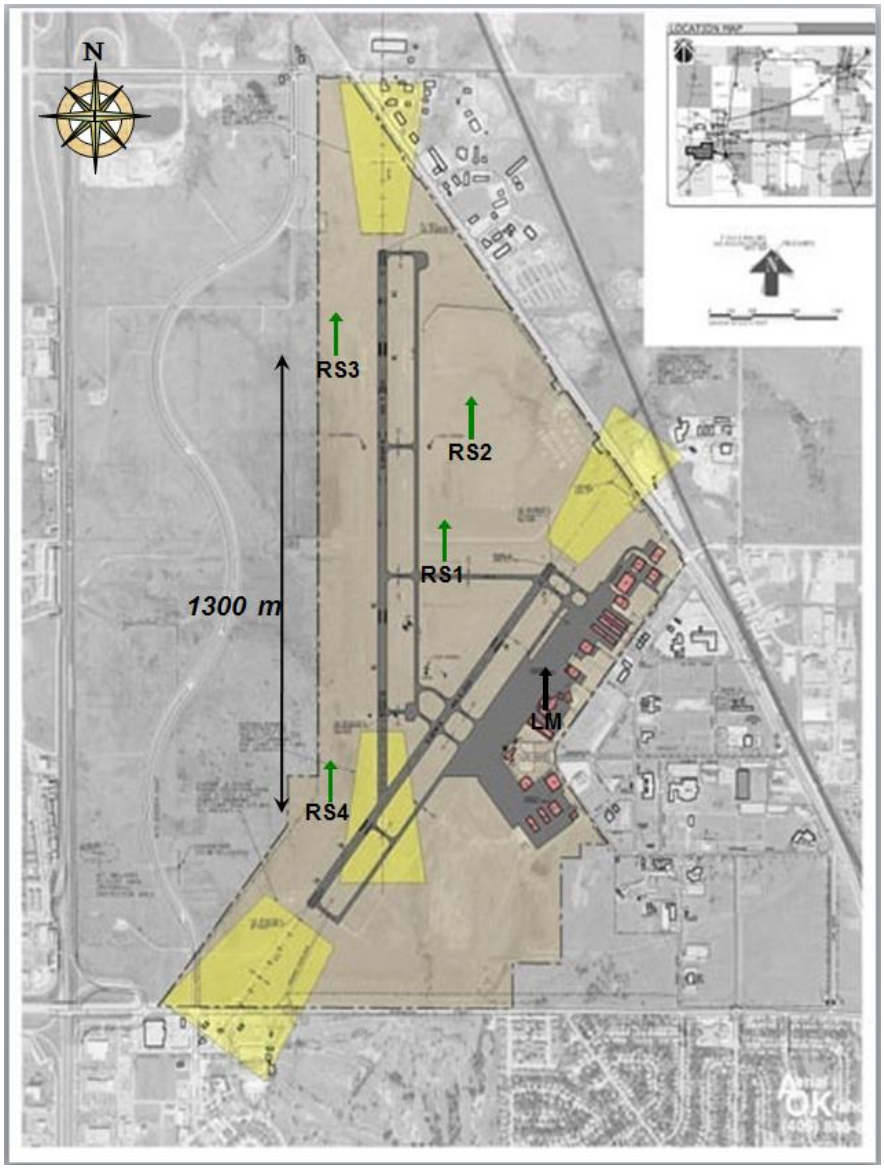

Figure 12 OU Reference Station Placement

3.1.2 Base Station

The OU LAAS Base station consists of the following

1.) Eight Freewave spread spectrum radios
2.) One central processing computer
    a. Window7 OS
    b. Quad core AMD Phenom II X4 @ 3.7 GHz
    c. 8GB RAM
    d. Two 8 port SeaLevel PCI-bus Serial Cards
3.) LAAS base station software (implemented in LabVIEW)

The LAAS base station (LBS) receives pseudorange information and navigation information (ephemeris, almanac, constellation health, etc.) from the reference stations and calculates the corresponding pseudorange corrections. The LBS then formats the pseudorange corrections and corresponding integrity information according to the "Augmentation System (LAAS) Signal-in-Space Interface Control Document" into the TYPE 1 LAAS Message. The LBS also constructs a TYPE 2 LAAS message. The TYPE 2 LAAS Message identifies the exact location for which the differential corrections provided by the LBS are referenced. The message also contains configuration data and data to compute a tropospheric correction [21]. Finally the LBS constructs a TYPE 4 message. The TYPE 4 message contains one or more data sets that each contain final approach data and associated vertical and lateral alert limits.
The ground station then transmits these messages to the airplane via a VHF data broadcast (VDB) transmitter.

3.1.3 VHF Data Broadcast (VDB)

The test system utilizes the following components for the VHF data broadcast.

1.) VDB transmitter (see Figure 13)
    a. Frequency range: 108.025 – 117.950MHz.
    b. Rated power: 10W to 80W
2.) 24v dc power supply
3.) VHF broadcast antenna
4.) GPS 1 PPS (used for constructing 500ms frames of the TDMA broadcast transmitted by the VDB)



Figure 13 Telerad (EM9009 A) VDB Transmitter

The LAAS Message will be sent to the VBD Transmitter from the LAAS base station via an RS-485 point-to-point asynchronous interface. The data packets from the LAAS base station will be formatted as shown in Table 5 [22].

Section	Size (bytes)	Description
Packet Sync	2	0x00FF(0x00FF followed by 0x0000)
Message ID	1	0x5C (see Table 6)
Message Length	1	Number of bytes in the application data section
Application data (optional)	0-255	Each byte can be any 8-bit value (One LAAS Message: Type 1, 2, or 4)
Checksum	2	16 bit additive checksum (LSB) first

Table 5 VDB Data Message Format

Message ID	Message Description	Size (bytes)	Application Data Format
0x5C	CAT-I Message	255x	Byte1- Reserved (set to 0x00) Byte2-Reserved(set to 0xFF) Byte3- TDMA Slot # (0-15) Bytes (4-255) LAAS Message (Type 1, 2, or 4)

Table 6 VDB Application Data Format

Each message that is received from the LAAS base station will be evaluated to determine if the VBD can transmit the message within the requested TDMA time slot and then either be discarded or transmitted at the configured frequency and power. A full explanation of the TDMA time slot evaluation can be found in [22]. The University of Oklahoma LAAS research facility transmits all LAAS message at a frequency of 113.55MHz and a power rating of 40 watts with an Effective Radiated Power of 120 watts.

3.1.4 Airborne Unit

The Airborne Unit receives the standard GPS signal, along with the LAAS information from the ground station, and combines them to provide precision position information to guide the plane along the approach path.

The Airborne Unit is composed of the following: a Control Unit, a Rockwell Collins GNLU 930 Multi-Mode Receiver (MMR), a Course Deviation Indicator (CDI) and two antennas. One antenna receives the GPS signal from the GPS constellation. The second antenna utilized is a standard VHF navigational antenna, which receives the LAAS signal that is transmitted by the VHF Data Broadcast (VDB) from the ground station. The functional layout of the airborne navigation system is illustrated in Figure 14.

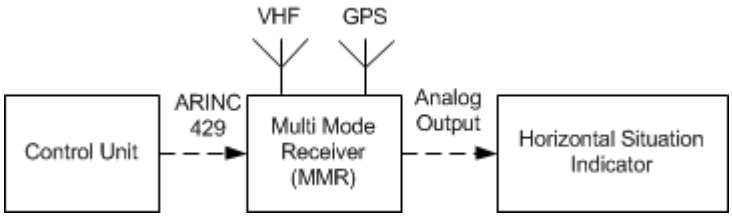

Figure 14 LAAS Airborne Unit

The Control Unit allows the pilot to select the frequency that is being utilized by the VDB to transmit the LAAS messages for the desired airport. Additionally the pilot can select which Final approach segment the MMR should use to provide guidance information. This information is found on the LAAS approach plate for the airport.

The Multi-Mode Receiver (MMR), shown in Figure 15, is designed to receive several different navigational signals, such as the GPS, LAAS, ILS, VOR, DME, and localizer. In the LAAS application, the MMR receives the standard GPS signal and the

LAAS VHF Data Broadcast (VDB). The MMR uses the pseudorange corrections, from the TYPE 1 LAAS message transmitted by the LAAS base station, to augment the GPS signal and provide a more accurate position calculation.

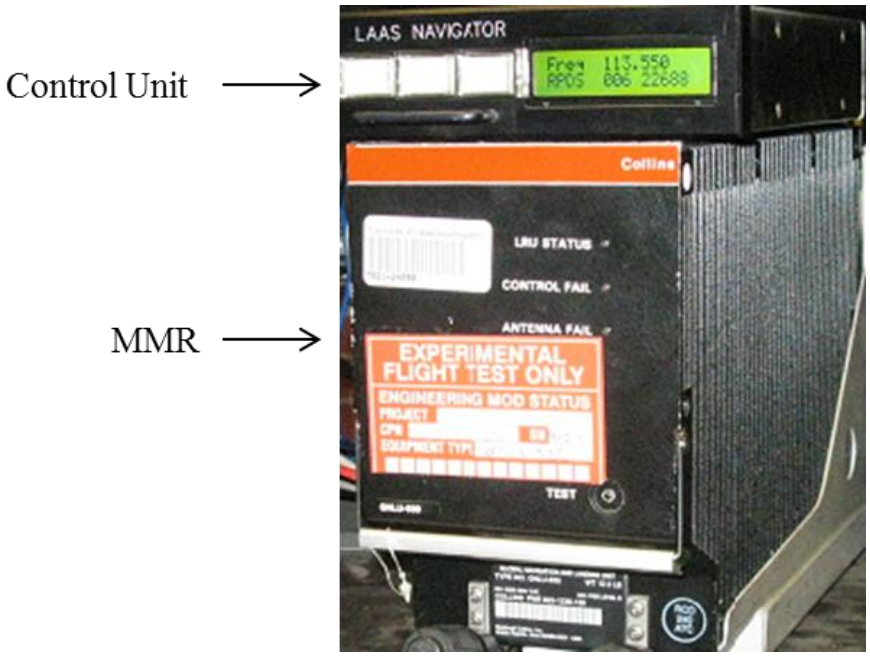

Figure 15 Rockwell Collins (GNLU-930) Multi-Mode Receiver (MMR)

The MMR also retrieves which final approach information will be used from the LAAS Type 4 message transmitted by the LAAS base station. The MMR compares this final approach information with the calculated position of the aircraft and determines how to navigate the plane onto the final approach path. The MMR relays this horizontal and vertical guidance information to the pilot through the airplane's navigation instruments, such as the Course Deviation Indicator (CDI) seen in Figure 16.



Figure 16 Course Deviation Indicator (CDI)

3.2 Closed-Loop Architecture

The block diagram in Figure 17 outlines the architecture of the new and novel real-time closed-loop LAAS system that was constructed to perform this research. This closed-loop system, unlike current LAAS implementations, utilizes a Local Monitor (LM) to calculate system error in real-time by validating ranging corrections that will be transmitted to airborne users. Utilizing the calculated system error an adaptive integrity monitor dynamically changes the broadcast integrity parameter σ_{pr_gnd}.

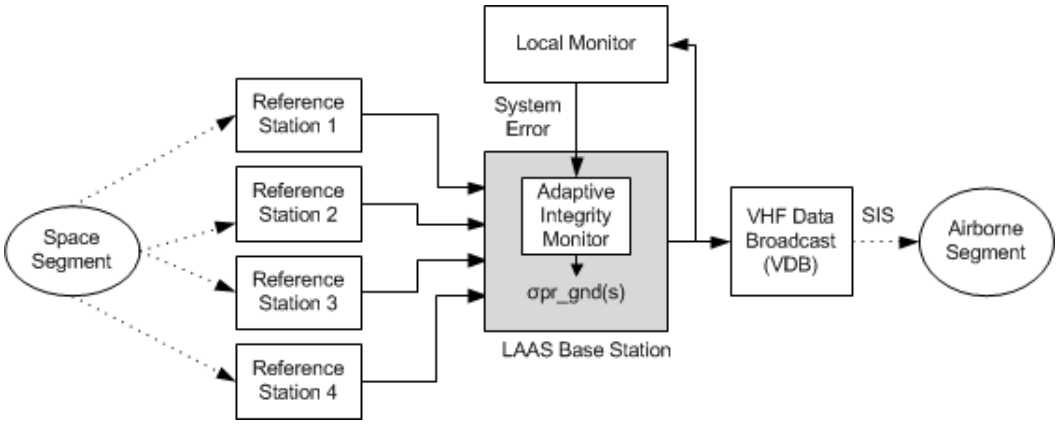

Figure 17 Closed-Loop Architecture

3.2.1 Local Monitor

The Local Monitor is composed of the following: a GPS antenna, a GPS receiver previously shown in Figure 10, and a Software Navigator. A block diagram of this architecture can be seen in Figure 18.

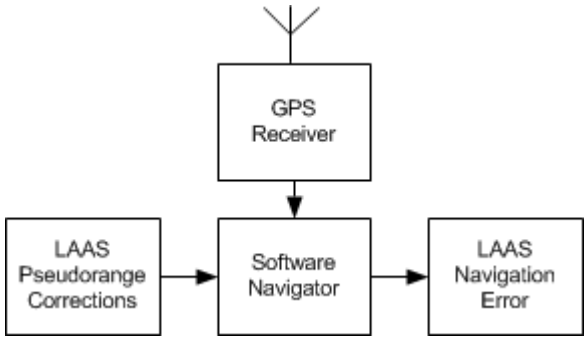

Figure 18 LAAS Local Monitor

The Local Monitor is a software based LAAS navigator whose actual position is at a known (surveyed) location. The LM calculates its position using a least squares single point solution as described in section 3.2.2 following. The LM uses the calculated position and the pseudorange corrections that are intended for transmission to the airborne unit to calculate its LAAS corrected position. The LAAS corrected position can then be compared to the known (surveyed) position of the LM. The difference in the two positions is representative of the actual accumulated error and is a real-time representation of the total LAAS ground station error. This error can now be used to deterministically modify the integrity broadcast parameters, specifically σ_{pr_gnd}, that will be transmitted to the airborne unit.

3.2.2 Local Monitor Position Calculation

According to the LAAS Minimum Aviation System Performance Standards the following algorithms for the computation of position will be implemented by the airborne navigation subsystem.

$$\Delta y = G\Delta x + \varepsilon \qquad \text{Eq. 11}$$

where:

Δy — *N* dimensional vector containing the differentially corrected pseudorange measurements minus the expected ranging values based on the location of the satellites and the location of the user (***x***)
Where N is the number of satellites

Δx — true four dimensional position/clock vector
[ground-track (x-axis), cross-track (y-axis), up (z-axis), (in a standard right-handed coordinate system) and clock] relative to the position/clock vector ***x*** for which the linearization has been made;

G — Geometry matrix consisting of *N* rows of line of sight vectors from each satellite to x, augmented by a "1" for the clock.

$G_i = [-\cos\theta_i \cos Az_i \quad -\cos\theta_i \sin Az_i \quad -\sin\theta_i \quad 1]$ = ith row of G

ε — is an *N* dimensional vector containing the errors in ***y***;

The weighted least squares estimate ($\Delta\hat{x}$) of the states can be found by:

$$\Delta\hat{x} = S\Delta y \qquad \text{Eq. 12}$$

where:

$$S = (G^T W^{-1} G)^{-1} G^T W^{-1} \qquad \text{Eq. 13}$$

$$W^{-1} = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_N^2 \end{bmatrix} \qquad \text{Eq. 14}$$

where:

σ_i^2 represents the total error as described in equation 5

As can be seen in equation 14 the weighting matrix W^{-1} is composed of σ_{pr_gnd} from the LBS and other error parameters ($\sigma_{iono,}$ $\sigma_{tropo, and}$ σ_{air}) calculated in the airborne unit [24]. These parameters that are specific to the airborne unit are set to zero, which leaves σ_{pr_gnd} as the remaining error parameter. If we assume an error free LAAS Ground Facility then we would set σ_{pr_gnd} to 0 as well; the resulting solution would then be a single-point least squares solution as described in equation 15 below [25].

$$\Delta\hat{x} = (G^T G)^{-1} G^T \Delta y \qquad \text{Eq. 15}$$

The Local Monitor implements equation 15 to calculate its position then applies the LAAS pseudorange corrections to calculate the final LAAS corrected position.

The LAAS corrected position is then compared to the known (surveyed) position of the LM to calculate the vertical and horizontal error.

3.2.3 Adaptive Integrity Monitor

The Adaptive Integrity Monitor is a new software based monitor that utilizes the calculated LAAS system error, specifically the vertical error, from the Local Monitor to deterministically set the values of σ_{pr_gnd} that are broadcast via the VDB.

The Adaptive Integrity Monitor reads the calculated vertical error from the Local Monitor and applies a smoothing filter, shown in equation 16, to generate a smoothed vertical error (VE_{sm}).

$$VE_{sm}[i] = \frac{1}{N}\sum_{j=0}^{N-1} VE[i+j] \quad \text{Eq. 16}$$

where:

VE_{sm}	The average vertical error over N samples.
N	The number of Vertical Error samples in the average.
VE	The instantaneous vertical error calculated by the Local Monitor (section 3.2.2)

A threshold algorithm is then applied to the smoothed vertical error ($VE_{sm)}$ to deterministically set $\sigma_{pr_gnd.}$ The threshold algorithm compares VE_{sm} to a range from 0 to 10m and sets the value of σ_{pr_gnd} according to the Table 7 following. All VE_{sm} values greater than 10 meters will be set to the .6. The range of σ_{pr_gnd} that is used in the threshold algorithm was determined via the sensitivity analysis explained in Chapter 4.

VE_{sm}	**σ_{pr_gnd}**
< 2m	.1
< 3m	.2
< 4m	.25
< 5m	.3
< 6m	.35
< 7m	.4
< 8m	.45
< 9m	.5
< 10m	.55
>10m	.6

Table 7 Threshold values

A flow chart of the Adaptive Integrity Monitor algorithm can be seen in Figure 19.

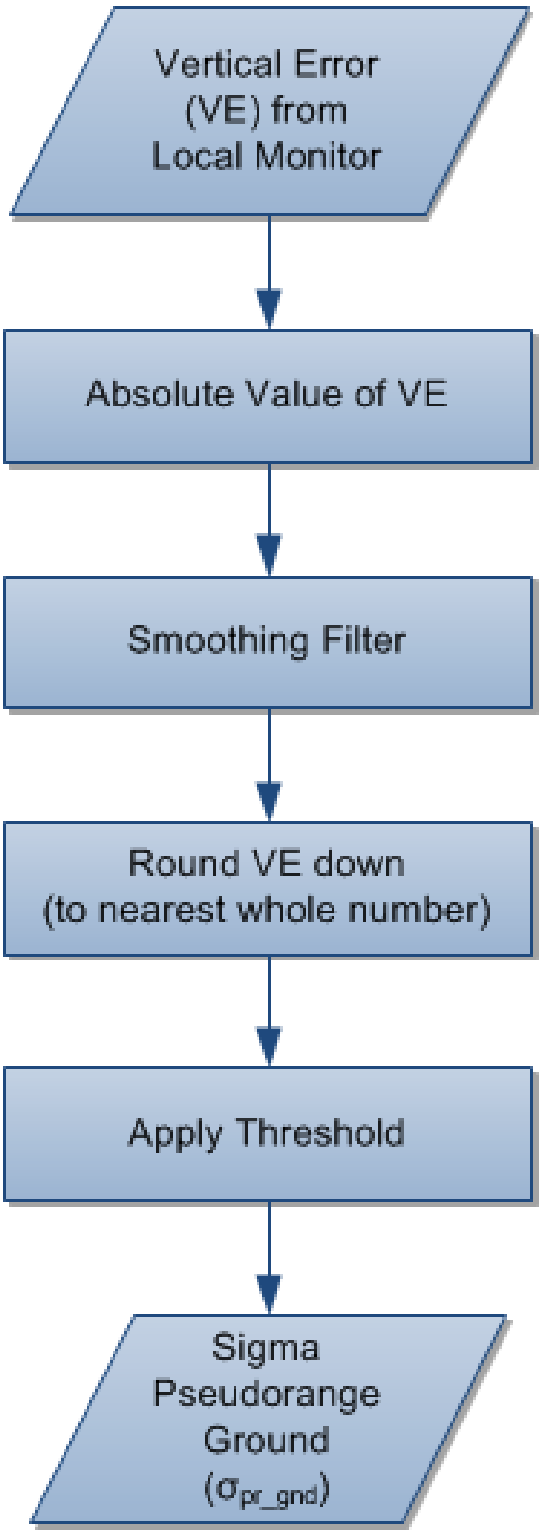

Figure 19 Adaptive Integrity Monitor Algorithm

Chapter 4 Experimental Results

4.1 Experimental Objective

All experiments described in this section were conducted utilizing the University of Oklahoma LAAS research facility described in Chapter 3.

One of the goals of this research is to determine the sensitivity of the LAAS airborne navigation solution to σ_{pr_gnd}. Based on the results of the sensitivity analysis described in this chapter, a closed loop adaptive integrity monitor was designed capable of deterministically assigning a value to σ_{pr_gnd} without reducing accuracy or overall system availability.

Sensitivity of σ_{pr_gnd} is determined with respect to availability of the LAAS system while monitoring Navigation System Error (NSE) to ensure system accuracy is not compromised.

Navigational System Error (NSE) is the difference in receiver-calculated position and the true position of the receiver. NSE can be divided into two components: Vertical Error (VE) and Horizontal Error (HE). The component of greatest concern is VE since it both contributes the most to NSE and, more importantly, it poses the largest threat to the user during precision approach and landing.

NSE exists in the position domain, and is typically reported from navigation devices in the form of Latitude, Longitude and Altitude. It is trivial to calculate the Vertical Component of NSE as can be seen in equation 17.

$$VE = VEcomputed - VEtruth \quad \text{Eq. 17}$$

To calculate the Horizontal Component of NSE a coordinate system conversion is required to allow HE to be reported in a more intuitive unit, meters, as opposed to

degrees, minutes, and seconds provided by the angular coordinates (latitude, longitude, altitude). The conversion from geodetic angular coordinates to geodetic Cartesian coordinates (X, Y, and Z) centered at the earth's center, is called Earth Centered Earth Fixed (ECEF) coordinates, is described in Appendix A.

Availability is defined as the ability of the navigation system to provide the required function and performance at the initiation of the intended operation. In the experiments and results described in this section availability is considered as the ability of the airborne navigator to provide a differentially corrected position solution. Two different methods were utilized to measure availability in this research and are described in following section.

4.2 Experimental Configuration

4.2.1 Dynamic Sensitivity Test

A dynamic sensitivity test was implemented to determine the sensitivity of the LAAS system availability as it is related to σ_{pr_gnd}. One primary goal of this test was to determine what value of σ_{pr_gnd} over-bound the Vertical and Horizontal Protection limits to the point that the system becomes unavailable.

For the purpose of this test, availability is determined by monitoring the state of the CDI glide slope flag on the airborne navigator. If the glide slope flag is displayed as seen in Figure 20 then the system is considered unavailable for the purpose of this test and for the purpose of CAT I or CAT II landing. If the glide slope flag is concealed, as seen in Figure 21, the system is considered available for approach and landing.



Figure 20 Navigation Unavailable

Glide Slope Flag



Figure 21 Navigation Available

The dynamic test utilized the following components:

1.) Four Reference Stations described in section 3.1.1

2.) LAAS base station described in section 3.1.2

3.) VHF Data Broadcast transmitter described in section 3.1.3

4.) Multi-Mode Receiver and Course Deviation Indicator described in section 3.1.4

5.) National Instruments myDAQ® data acquisition device

6.) Custom software test harness developed in LabVIEW® described below

A logical diagram of the test system can be seen in Figure 22.

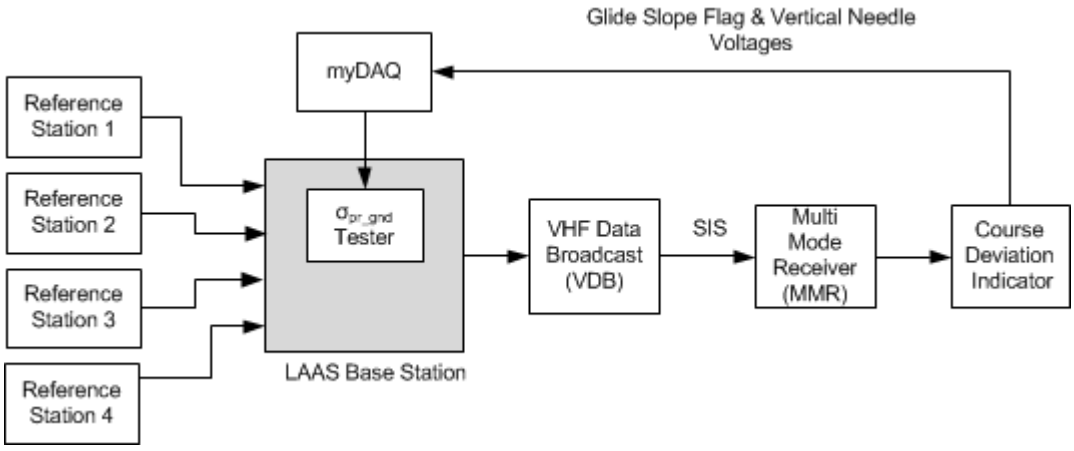

Figure 22 Dynamic Sensitivity Diagram

A flow chart describing the algorithm used to perform the sensitivity test can be seen in Figure 23.

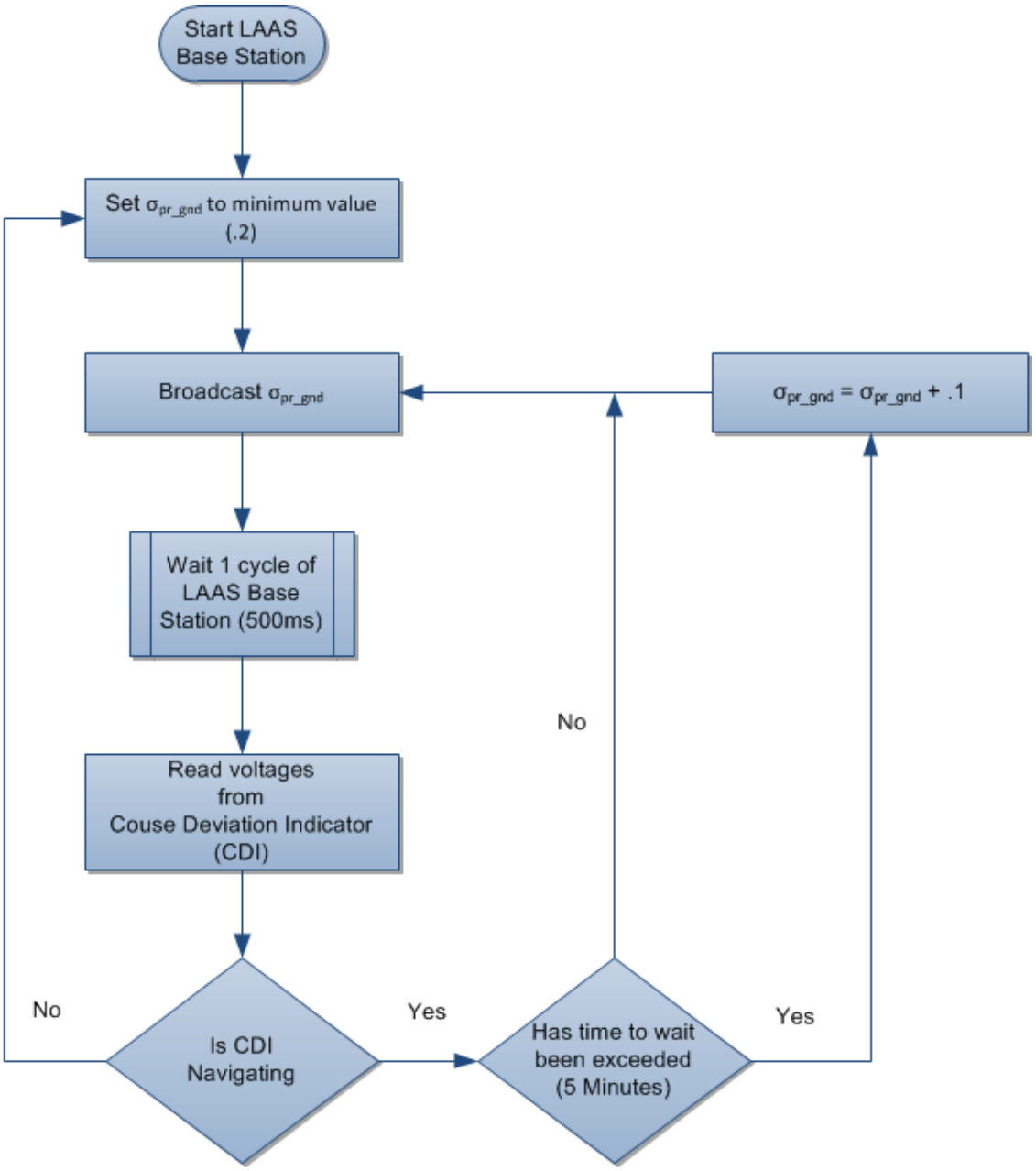

Figure 23 Dynamic Sensitivity Flowchart

During each iteration of the test, data was logged to the local hard drive. The logged data included a UTC timestamp, the broadcast value of σ_{pr_gnd}, CDI Glideslope Flag voltage, CDI vertical needle voltage, the LAAS corrected position, Vertical Error, Horizontal Error. Data was collected for 12 hours to match the approximate time it takes a single GPS satellite to orbit the earth. The 12 hour duration guarantees all satellites were in view at least one time during the course of the test. The results of this sensitivity test are shown in Figure 24 following. The stair-step pattern displayed in Figure 24 is a result of the incremental changes of σ_{pr_gnd} until the system becomes unavailble as outlined in the flowchart in Figure 23 previously. When the value of σ_{pr_gnd} makes a change from a larger value to .2 this is an indication the system has become unavailable, as shown in Figure 25. The data illustrates that the system is rarely available once the value of σ_{pr_gnd} exceeds a value of .6.

Figure 26 following illustrates the percentage of time a given value of σ_{pr_gnd} was broadcast during the 12 hour period of time the test was conducted. This figure further indicates that a significant loss of availabilty occurred once the value of .6 σ_{pr_gnd} was broadcast. Based on the analysis of this test .6 σ_{pr_gnd} was selected as the maximum value of σ_{pr_gnd} that will be broadcast from the test system.

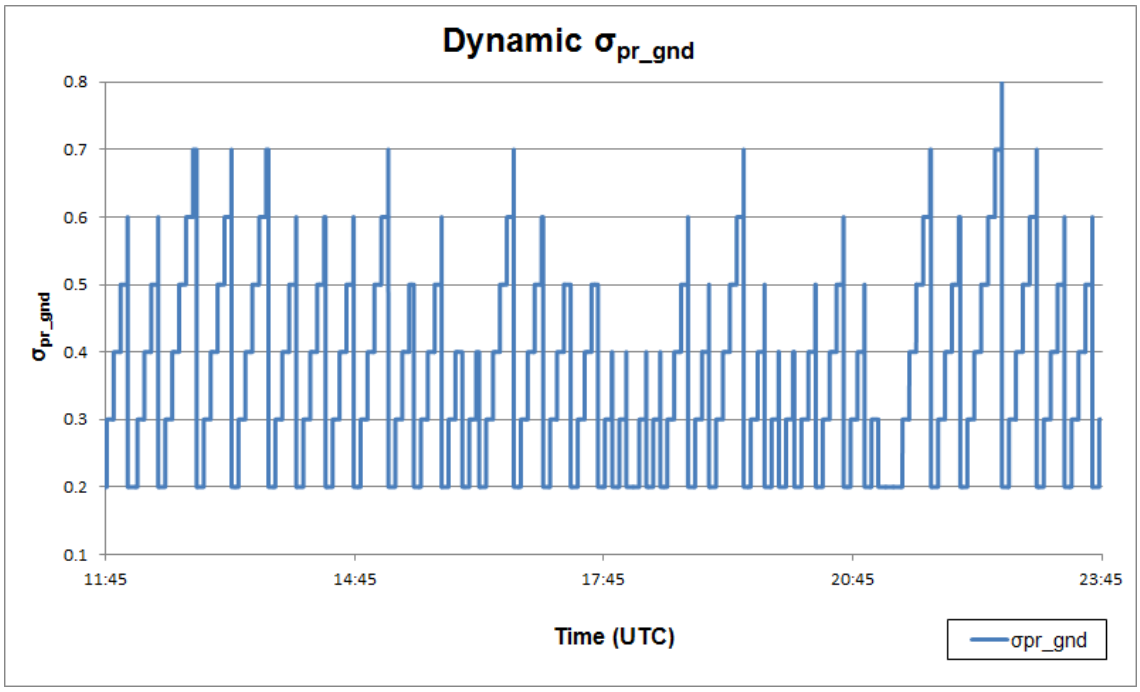

Figure 24 Dynamic σpr_gnd Sensitivity Analysis

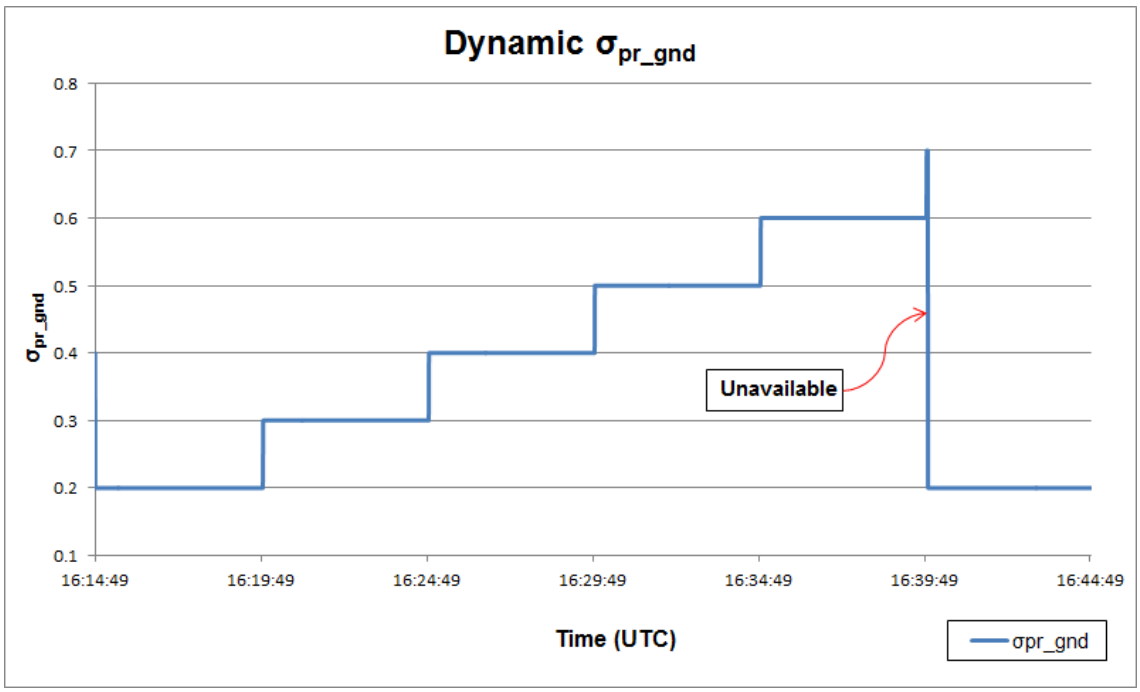

Figure 25 Loss of Availability

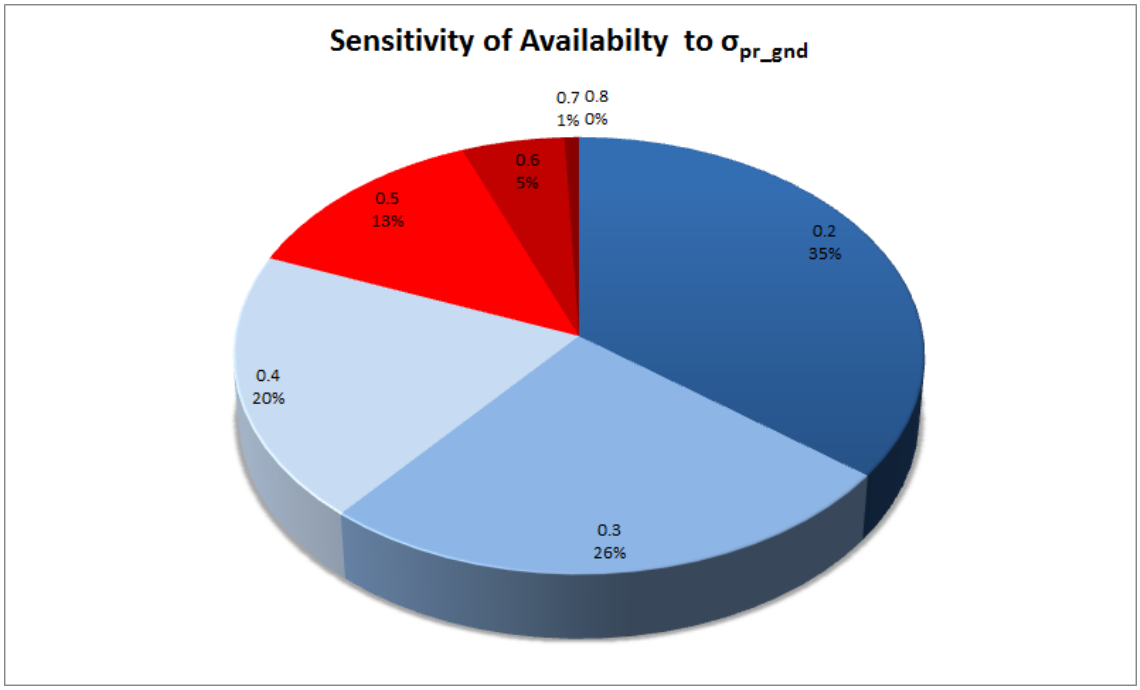

Figure 26 Availability shown as Percent σ_{pr_gnd}

4.2.2 Multi-Orbit Static Sensitivity Test

Utilizing the results of the dynamic sensitivity test, described in section 4.2.1, a set of multi-orbit static sensitivity tests were executed. The goal of these tests was to quantify the availability of the system at a specified σ_{pr_gnd} ,verify that the Navigation System Error stayed within acceptable bounds, and establish a baseline vertical error for static values of σ_{pr_gnd} that could be used for analysis of the new adaptive integrity monitor.

For the purpose of these tests, availability was determined by reading the differential mode indicator, present in ARINC label 273. ARINC label 273 is output on the ARINC 429 bus from the Airborne Unit's Multi-Mode Receiver. To retrieve the

differential mode indicator a custom logger was constructed. This ARINC logger is described in detail in section 4.3.1.

This test utilized the following components:

1) Four Reference Stations described in section 3.1.1
2) LAAS base station described in section 3.1.2
3) VHF Data Broadcast transmitter described in section 3.1.3
4) Multi-Mode Receiver section 3.1.4
5) A data logging computer
    a. Condor® CEI-715 PCMCIA card, shown in Appendix F
    b. Custom ARINC data logging software developed in C++

The sensitivity test was conducted as follows:

1.) Set a static value for σ_{pr_gnd}
2.) Start LAAS base station software
3.) Wait for LAAS base station to start calculating differential corrections
4.) Start VDB Transmitter
5.) Start ARINC data logger to collect MMR ARINC messages
6.) Collect ARINC messages from MMR for a minimum of 24 hours

This process was repeated for the following values of σ_{pr_gnd} (.1, .2, .4, and .6). Each test was run for a minimum of 24 hours to ensure two complete orbits of each GPS satellite. The data logging software generated two types of log files for each test.

One log file contains all ARINC labels in their raw format as output from the MMR. The second log file includes a subset of the ARINC messages that have been decoded and will be used for post processing. The decoded ARINC messages include

the following data UTC Timestamp, latitude, longitude, altitude, and navigation mode. The information collected in the decoded log file was used in post processing to calculate availability and Navigation System Error (NSE) for each static value of σ_{pr_gnd}. The results of these tests are discussed in section 4.4 following.

4.2.3 Real-Time Adaptive Integrity Monitor

Applying the σ_{pr_gnd} range determined from the sensitivity analysis, and utilizing the real-time vertical error calculated by the local monitor, the adaptive integrity monitor algorithm described in Figure 19 was implemented in LabVIEW® and tested. The goal of the test was to demonstrate that σ_{pr_gnd} could be deterministically modified in real-time without negatively impacting availability or accuracy of the system. This test utilized the following components:

1) Four Reference Stations described in section 3.1.1
2) LAAS base station described in section 3.1.2
3) Local Monitor described in section 3.2.1
4) Adaptive Integrity Monitor implemented in LabVIEW®
5) VHF Data Broadcast transmitter described in section 3.1.3
6) Multi-Mode Receiver section 3.1.4
7) A data logging computer
    a. Condor® CEI-715 PCMCIA card, shown in Appendix F
    b. Custom ARINC data logging software developed in C++

The experimental protocol proceeded as follows:

1.) Start LAAS base station software

    a. Local Monitor is started by base station

2.) Enable Adaptive Integrity Monitor

3.) Wait for LAAS base station to start calculating differential corrections

4.) Start VDB Transmitter

5.) Start ARINC data logger to collect MMR ARINC messages

6.) Collect ARINC messages from MMR for a minimum of 24 hours

The MMR's calculated user position and current navigation mode was collected using the custom ARINC data logger as described in the multi-orbit static sensitivity test described in section 4.2.2. This data was used in post processing to generate availability and Navigation System Error metrics that were used to validate the effectiveness of the new real-time closed-loop adaptive integrity monitor. These results are discussed in section 4.4.

4.3 Data Collection and Processing

4.3.1 Data Collection

The data used to build the algorithms and evaluate the performance of the adaptive integrity monitor were collected from three different sources:

1.) LAAS Base Station

2.) Multi-Mode Receiver

3.) Course Deviation Indicator

Figure 27 following illustrates all of these data collection points.

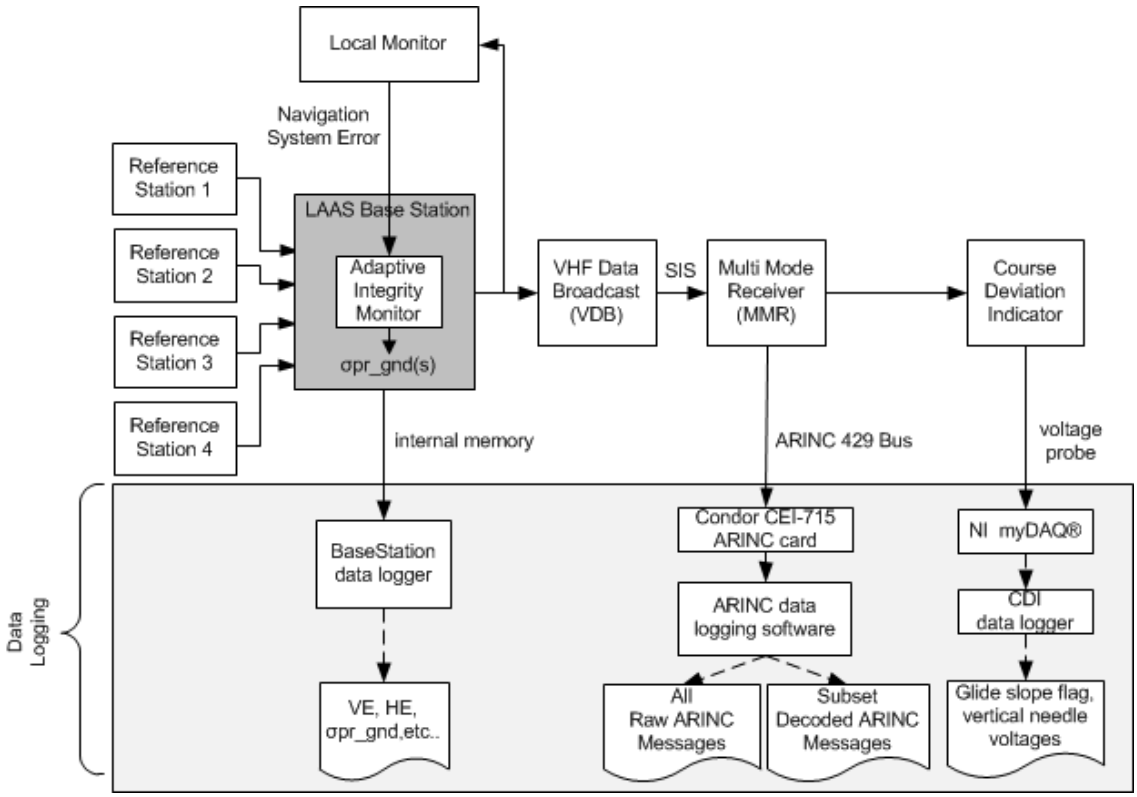

Figure 27 Data logging diagram

The data collected from the base station includes the vertical and horizontal navigation system error calculated by the local monitor, the σ_{pr_gnd} values that are broadcast from the base station, and a UTC time stamp. This data was collected at a rate of 2 Hz and written to a log file on the local hard drive of the base station computer.

The data collected from the Multi-Mode Receiver that was used for the experiments described above include the LAAS corrected user position calculated by the Multi-Mode Receiver, the navigation mode flag, and a UTC timestamp. This information was collected by custom data logging software written in C++ that interfaced to a Condor CEI-715 ARINC card to read and decode the desired ARINC labels output by the MMR on the MMR's ARINC 429 bus. The custom data logging software was written to generate two log files. One log file contains all of the raw ARINC labels transmitted by

the MMR. The second log file contains only a subset of the ARINC labels transmitted by the MMR, which are needed for the experiments described above. The ARINC labels listed in Table 8, following, are the labels that were used for the experiments described above. The labels were decoded by the data logger and written to the local hard drive of the data logging computer. All labels were decoded using the specifications outlined in Rockwell Collins Interface Control Document for the MMR GNLU 930[26]. The detailed label definitions used for decoding the ARINC labels listed in Table 8, below, can be found in Appendix C.

ARINC Label	Description
110	Latitude
111	Longitude
120	Latitude Fine
121	Longitude Fine
140	UTC Fine
150	UTC
273	GPS Sensor Op Mode (includes navigation mode indicator)
370	WGS Height in feet

Table 8 MMR ARINC Labels

The data collected from the Course Deviation Indicator included the voltages for the Glide Slope Flag and the Vertical Needle. A custom data logger was developed using LabVIEW®, to interface to a National Instruments myDAQ® to read the voltages, perform the analog to digital conversion, and log the voltages and state of the CDI to a file on the local hard drive of the data logging computer. The voltages collected were used to determine if the CDI was in differential navigation mode based on the state of the glide slope flag. If the glideslope flag was displayed, as shown in Figure 20, then the glideslope was considered unavailable, and the data logger application logged the state of the CDI as unavailable, otherwise the state was logged as available. Table 9, below, shows the voltage ranges of both the glide slope flag and the vertical needle.

A wiring diagram of the CDI used to design this data acquisition system can be found in Appendix D.

	Voltage Range	Description
Glide Slope Flag	0v to .7v	.7v flag is concealed (glideslope available) 0v flag is displayed (glideslope unavailable)
Vertical Needle	-1.5v to 1.5v	0v vertical needle is centered -1.5v vertical needle is (fully deflected upwards indicating unavailable)

Table 9 CDI Voltage Ranges

4.3.2 Data Processing

For the Multi-Orbit Sensitivity tests and the performance evaluation of the Adaptive Integrity Monitor a minimum of 24 hours of data was collected and then post-

processed to calculate vertical error, standard deviation of vertical error, and system availability.

To calculate these metrics a post-processing application was written in C#. The post-processing application combined multiple log files that were generated for a single test, down-sampled the data to 1Hz, and performed the necessary calculations to generate vertical error and determine if the MMR was in differential navigation mode. The vertical error was determined using known altitude of the MMR and the calculated user position output by the MMR (see equation 17). The new down-sampled data was then written to a new comma separated value (.csv) file. A screen shot and main processing method of the post-processing application can be seen in Appendix E.

Finally, Microsoft Excel was used to generate vertical error graphs, calculate the standard deviation of the vertical error, and calculate the % availability of the system based on the amount of time the MMR was in differential navigation mode. To calculate % availability the following equation was used:

$$\%\ Available = \frac{(X-Y)}{Y} \times 100 \qquad \text{Eq. 18}$$

where:

X Duration of test in seconds

Y Total # of seconds the MMR was not in differential navigation mode

4.4 Experimental Results

4.4.1 Multi-Orbit Static Sensitivity

The analysis of the test data collected for the multi-orbit sensitivity test described in section 4.2.2, validated that as σ_{pr_gnd} increased, availability of the system decreased as a result of the over-bounding of the vertical and horizontal protection levels induced by inflated values of σ_{pr_gnd}.

The following four figures represent the vertical error for each of the multi-orbit sensitivity tests that were performed. The data in these figures show that less than a meter of vertical error can be maintained provided the system is not over-bound, causing loss of system availability. Furthermore, it demonstrates that a value of .6 for σ_{pr_gnd} is sufficiently large to over-bound the system and cause a degradation in position accuracy due to the loss of availability, as was initially indicated in the dynamic sensitivity test previously described in section 4.2.1. Table 10, following, shows the availability of the system associated with each Multi-Orbit test that was performed.

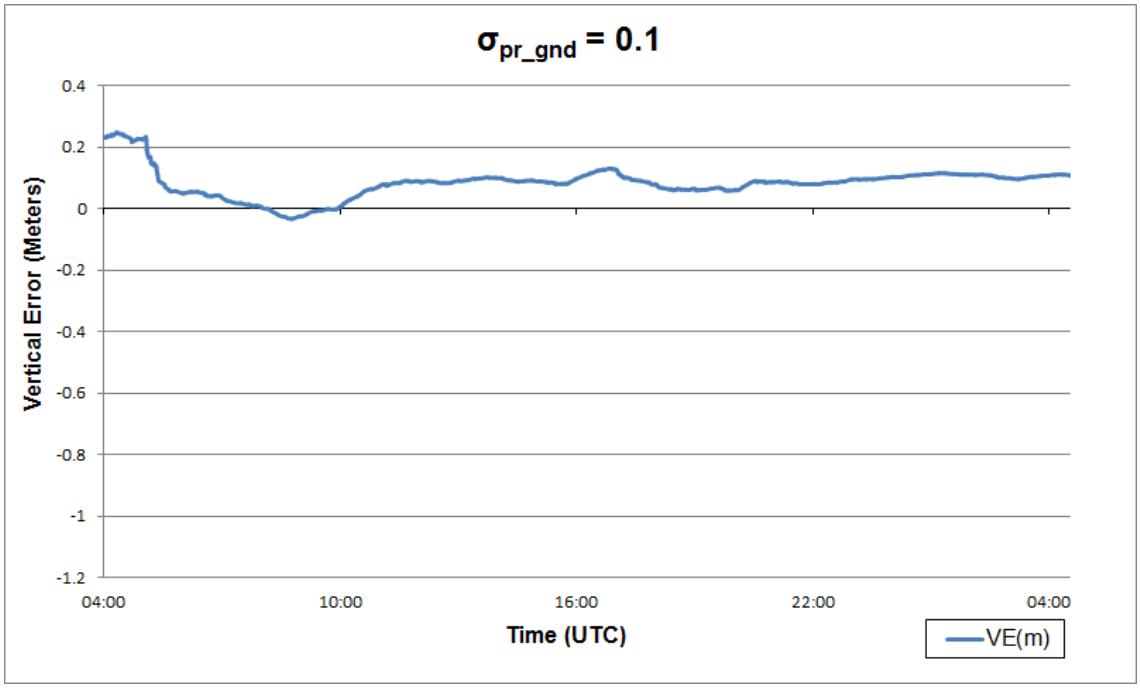

Figure 28 Vertical Error for Multi-Orbit Sensitivity Test with σ_{pr_gnd} = 0.1

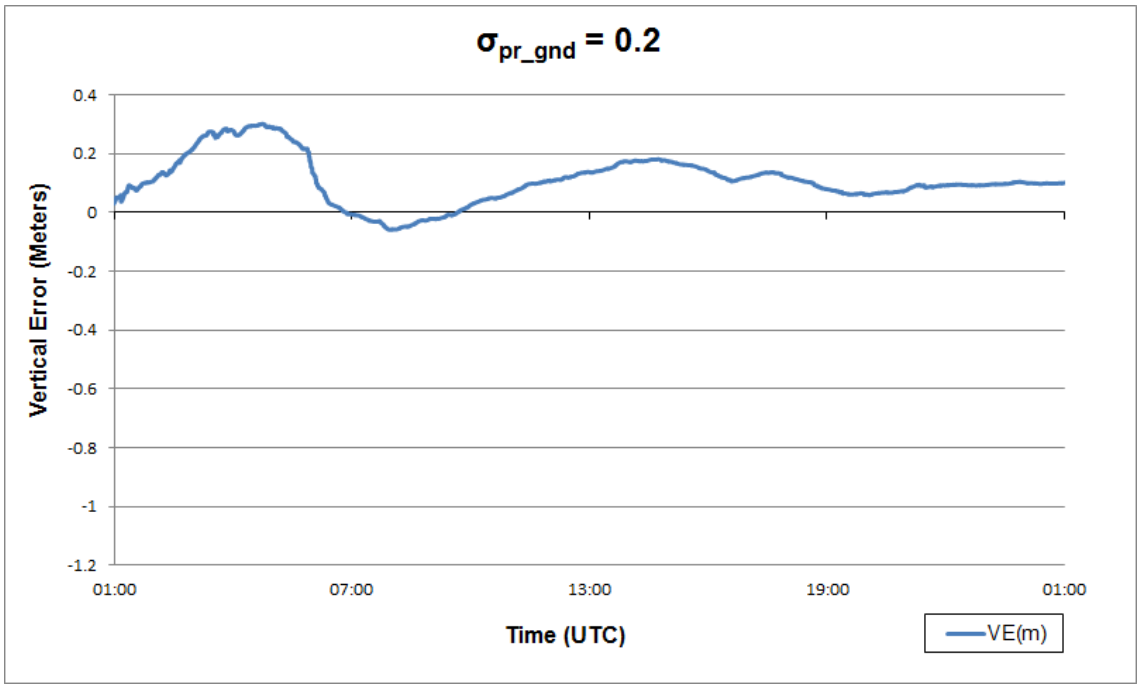

Figure 29 Vertical Error for Multi-Orbit Sensitivity Test with σ_{pr_gnd} = 0.2

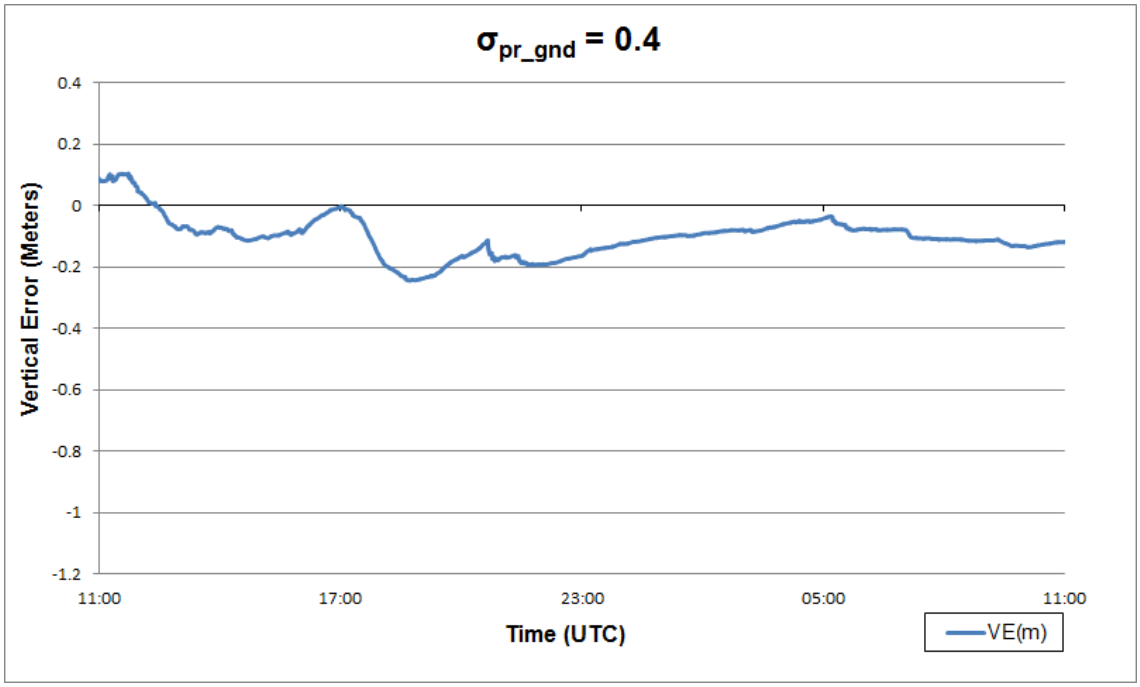

Figure 30 Vertical Error for Multi-Orbit Sensitivity Test with σ_{pr_gnd} = 0.4

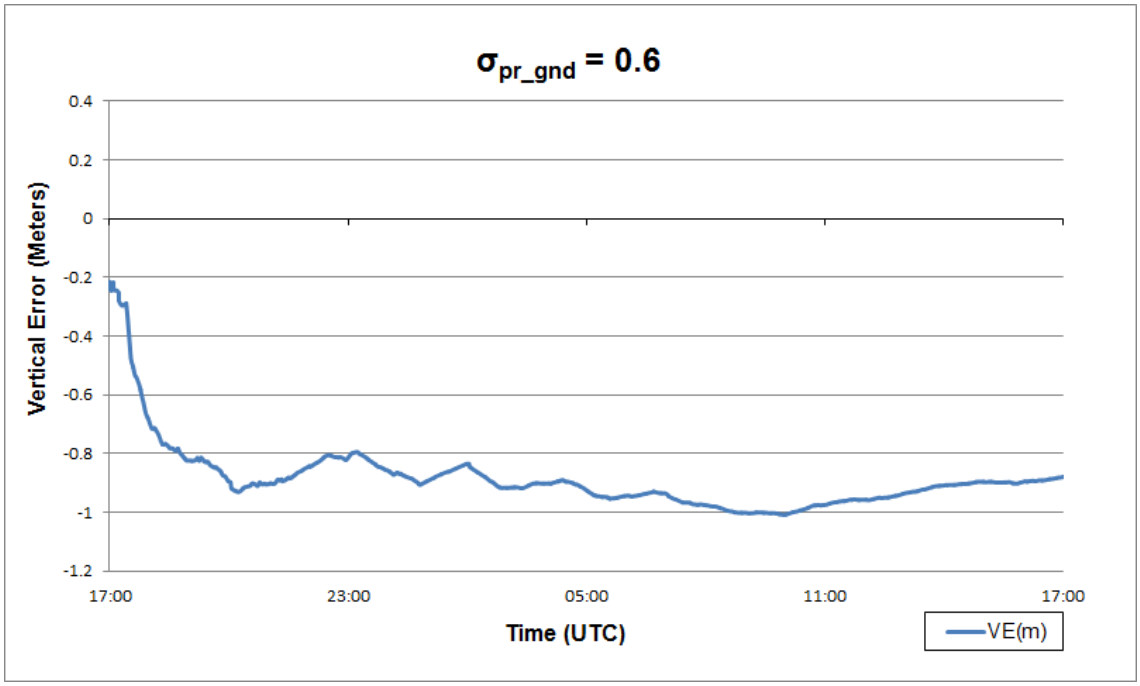

Figure 31 Vertical Error for Multi-Orbit Sensitivity Test with σ_{pr_gnd} = 0.6

Duration (h)	σ_{pr_gnd}	% Availability	$\left\|\overline{VE}\right\|\ (m)$
24	.1	99.76	.1336
24	.2	99.52	.1188
24	.4	93.95	.1203
24	.6	32.69	.8707

Table 10 Multi-Orbit Availability with static σ_{pr_gnd}

4.4.2 Real-Time Adaptive Integrity Monitor

The final design of the Adaptive Integrity Monitor described in section 4.2.3 includes a smoothing filter designed to reduce large variations in vertical error over short time periods that would result in large changes of σ_{pr_gnd} in a short period time. This design is a result of initial testing that was performed without a smoothing filter. The vertical error of this test is shown in Figure 32 following. The vertical error is clearly within the 2.9m acceptable vertical accuracy requirement for GBAS Service Level F which is required for CAT IIIb precision approach. However, the recorded system availability of 98.64% was less than desirable, as shown in Table 11 following.

It was hypothesized that the Airborne Navigation position algorithms of the MMR were not designed to account for rapid changes in σ_{pr_gnd}. This was based on the performance metrics collected from the static sensitivity tests indicating that higher level of availability was achievable with this test system.

In an effort to validate this theory a smoothing filter was added to the Adaptive Integrity Monitor and the tests were executed again.

The results of this second test, that included a 120 point smoothing filter, validated the hypotheses. Figure 33, following, demonstrates the vertical error of this test remained within the acceptable limits for GBAS Service Level F. Additionally, the

results of this second test demonstrated a system availability improvement to 99.91%, as shown in Table 11 following.

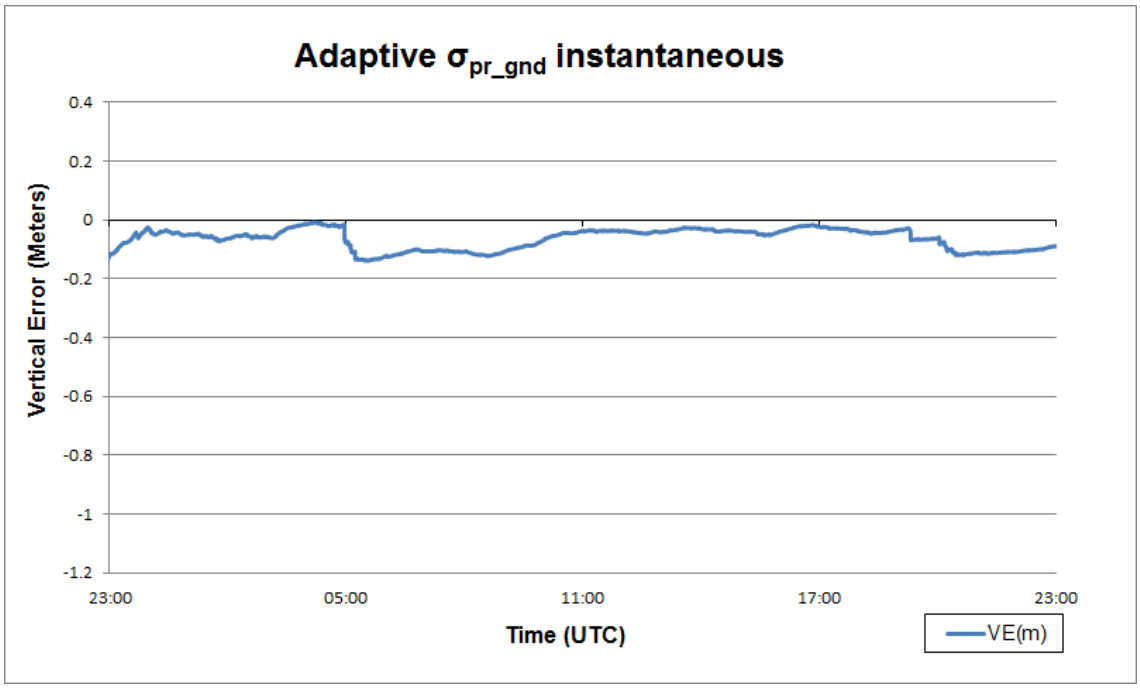

Figure 32 Adaptive σ_{pr_gnd} Instantaneous

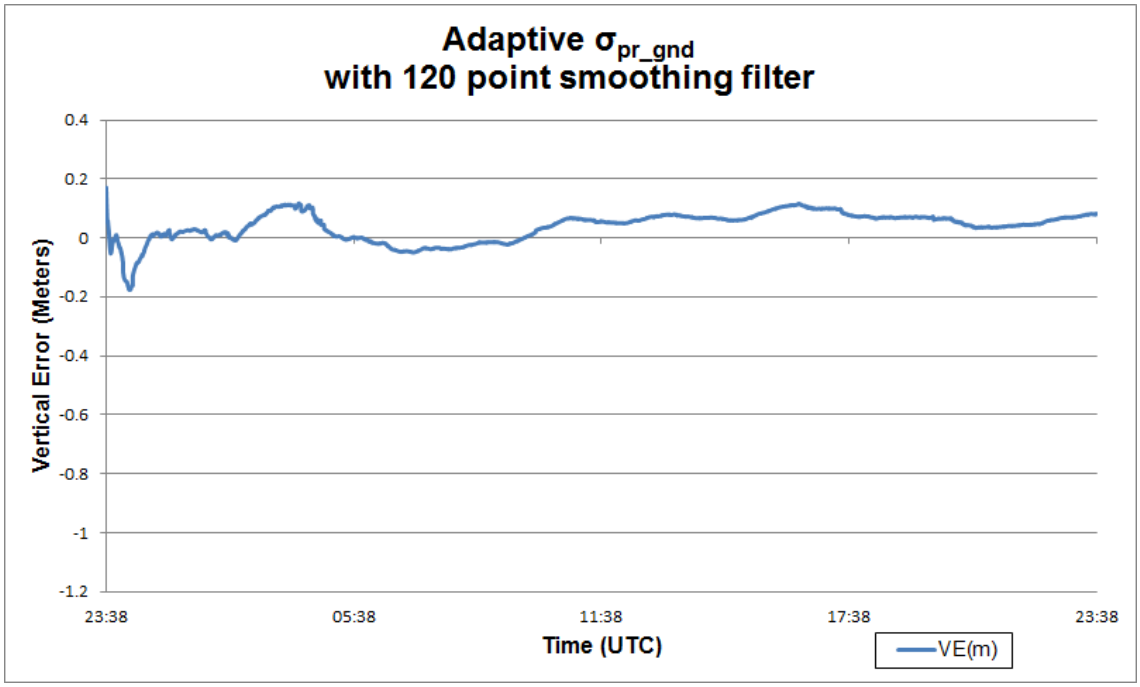

Figure 33 Adaptive σ_{pr_gnd} with a Smoothing Filter

4.5 Results Summary

Sensitivity analysis was performed over a range of σ_{pr_gnd} values to characterize accuracy and availability. The results of this sensitivity analysis were used to design the new Adaptive Integrity Monitor and verify the theoretical advantages.

The data in Table 11 supports the theoretical advantages of the Adaptive Integrity Monitor over the current practice of probabilistic determination and over-bounding of σ_{pr_gnd}. Using the closed loop Adaptive Integrity Monitor the broadcast parameter σ_{pr_gnd} can be dynamically changed avoiding over-bounding of the system and therefore increasing system availability without degradation in accuracy.

Duration(h)	σ_{pr_gnd}	% Availability	$\left\|\overline{VE}\right\|\ (m)$	$\sigma_{VE}\ (m)$
24	.1	99.76	.1336	.1560
24	.2	99.52	.1188	.1272
24	.4	93.95	.1203	.1001
24	.6	32.69	.8707	.1569
24	Variable-Instantaneous	98.64	.1153	.1799
24	Variable-Smoothed	99.91	.0563	.0485

Table 11 Test Results Summary

Chapter 5 Conclusions and Future Research

The research described in this dissertation presents a new and novel set of algorithms and architecture for a real-time adaptive integrity monitor that utilizes navigation system vertical error to deterministically set the broadcast integrity parameter $\sigma_{pr_gnd.}$

This deterministic evaluation of σ_{pr_gnd} as a function of navigation system vertical error eliminates the need for probabilistic over-bounding of. $\sigma_{pr_gnd.}$ By eliminating the need to over-bound integrity parameters an increase of system integrity was achieved. A subsequent result of increasing integrity was an increase in system availability.

The data collected in this research validated that the new closed loop integrity monitor is capable of improving system availability to 99.9%. This level of availability can assist LAAS in meeting its final goal of providing differential navigation suitable for Category II and Category III instrument landings.

Additionally, the research described in the dissertation outlines a new method for real-time analysis of airborne navigation algorithm sensitivity to $\sigma_{pr_gnd.}$ This sensitivity analysis was fundamental to the design and performance evaluation of the new adaptive integrity monitor.

Finally this dissertation presented a complete and functional real-time implementation of a Local Area Augmentation System that utilizes closed-loop feedback to improve integrity and availability.

5.1 Recommendations for Future Research

If LAAS is to meet its final goal of providing differential navigation suitable for Category II and Category III instrument landings additional improvements in integrity and availability will be required. The research described in this dissertation provides a set of algorithms and concepts that provide a foundation that can be expanded on to realize this goal.

The expansion of these concepts might include flight tests to further evaluate the improvements in integrity and availability due to the new adaptive integrity monitor.

Future work should also include a sensitivity analysis of the system availability related to the sample size of the smoothing algorithm in the new adaptive integrity monitor. The goal of this analysis would be to determine the ideal sample size for the smoothing algorithm that would achieve the greatest availability improvement.

The sensitivity research could also include testing the adaptive integrity monitor with a variety of different airborne navigation units to investigate any dependencies between measured performance improvements and the airborne navigator utilized for this research.

5.2 Summary

The real-time adaptive integrity monitor using closed loop feedback that was presented in this dissertation is a unique method for deterministically assessing the error of the LAAS ground subsystem, in contrast to the current probabilistic models used today. The research demonstrates that the adaptive integrity monitor can provide performance improvements in integrity and availability of the LAAS Signal-In-Space.

References

[[1] Juan Blanch, Todd Walter, and Per Enge, “Satellite Navigation for Aviation in 2025” Proceedings of the IEEE, vol. 100, pp. 1821-1830, 2012.

[2] A. El-Rabbany, Introduction to GPS: the Global Positioning System, Norwood, MA, Artech House, 2nd ed., 2006.

[3] “Minimum Aviation System Performance Standards for the Local Area Augmentation System,” Document No. RTCA/DO-245, 1998

[4] C. Davis, J. Dyer, A. Archinal, H. Wen, and J. Fagan, “Conceptualization and Implementation of a Closed-Loop Ground Based Augmentation System,” Proceedings of the 20th International Technical Meeting of the Satellite Division of the Institute of Navigation ION GNSS, Fort Worth, Texas, pp 461-466. September 25-28, 2007.

[5] C. Hegarty and E. Chatre, “Evolution of the Global Navigation Satellite System (GNSS),” Proceedings of the IEEE, vol. 96, no. 12, pp. 1902-1917, 2008.

[6]C. Davis, R. Pendergraft, J. Henderson, J. Dyer, M. Yeary, and J. Fagan, “Architecture and Performance of an Instrumented RF System that Utilizes the GNSS Satellite Network,”2013 IEEE International Instrumentation and Measurement Technology Conference, Minneapolis, Minnesota, May 6-9, 2013.

[7] Blackwell, Earl G., "Overview of Differential GPS Methods", *NAVIGATION*, Vol. 32, No. 2, Summer 1985, pp. 114-125.

[8]FAA-AJW44, “Non-Fed Specification FAA-E-AJW44-2937A Category I Local Area Augmentation System Ground Facility,” tech. rep., United States Department of Transportation Federal Aviation Administration, October 2005.

[9]T.Murphy and T. Imrich, “Implementation and operational use of ground-based augmentation systems (GBASs) A component of the future air traffic management system,” Proceedings IEEE, vol. 96, no. 12, pp. 1936-1957, Dec 2008.

[10]FAA GPS Product Team, “Global Positioning System (GPS) Standard Positioning Service (SPS) Performance Analysis Report”, January 2012.

[11] “Minimum Operational Performance Standards for GPS Local Area Augmentation System Airborne Equipment”, RTCA SC-159 DO-253C, 16 Dec. 2008

[12] Seo J, Lee J, Pullen S, Enge P, Close S."Targeted Parameter Inflation Within Ground-Based Augmentation Systems To Minimize Anomalous Ionospheric Impact." Journal Of Aircraft, vol 49, no. 2, pp 587-599. March-April 2012.

[13] P.B. Ober, “LAAS Integrity the Bayesian Way,” Proceedings of the 2000 *National Technical Meeting of the Institute of Navigation*, Anaheim, CA, pp 236-245

[14]B. Pervan, S. Pullen, and I. Sayim, “Sigma Estimation, Inflation, and Monitoring in the LAAS Ground System.” ION GPS 2000, Salt Lake City, UT, pp 1234-1244, September 2000.

[15] Braff, Ronald. "Description of the FAA's local area augmentation system (LAAS)." Navigation 44.4 (1997): 411-423.

[16] Pervan, B., and I. Sayim. "Issues and Results Concerning the LAAS gnd pr _ σ Overbound." IEEE PLANS. 2000

[17] Sayim, Irfan, et al. "Experimental and theoretical results on the LAAS sigma overbound." Institute of Navigation: Proceedings of ION GPS-2002, Portland, Oregon (2002): 1916-1924.

[18] “Global Positioning System Standard Positioning Service Performance Standard (SPS)”, 4th edition, September 2008.

[19] Lee, J., Pullen, S., Xie, G., & Enge, P. (2001, June). LAAS sigma-mean monitor analysis and failure-test verification. In Proceedings of the ION 57th Annual Meeting (pp. 11-13).

[20] Dyer, John William. *A New Siting Model for Reference Station Placement in a Ground Based Augmentation System*. PhD thesis, The University of Oklahoma, 2008.

[21] Area, GNSS-Based Precision Approach Local. "Augmentation System (LAAS) Signal-in-Space Interface Control Document (ICD)." Washington, DC, RTCA SC-159, WG-4, DO-246D (2008).

[22] “VDB Series 90x9 GBAS Interface Control Document (ICD).” Telerad Navigation, Version 1-6 (2003)

[23] Wullschleger, V., R. Braff, and T. Urda. "FAA LAAS Specification: Requirements for Performance Type." In European Institutes of Navigation GNSS 2000 Conference. 2000.

[24] Davis, Chad. *Conceptualization and Implementation of a New and Novel Ground Based Augmentation System Utilizing Feedback*. PhD thesis, The University of Oklahoma, 2007.

[25] Axelrad, P., and R. G. Brown. "GPS navigation algorithms." Global Positioning System: Theory and applications. 1 (1996): 409-433.

[26] “Interface Control Document (ICD) for the Analog MMR GNLU (920/930/940/950)”, Revision A, Rockwell Collins, (2001).

[27] B. Ho_man-Wellenhof, H. Lichtenegger, and J. Collins, GPS: Theory and Practice. New York: Springer-Verlag, 5th ed., 2001.

Appendix A: Conversion from LLA to ECEF

The conversion from Latitude, Longitude, and Altitude (LLA) to Earth Centered Earth Fixed (ECEF) in meters can be performed using the following equations.

$$X = (N + h)\cos\varphi\cos\lambda$$

$$Y = (N + h)\cos\varphi\sin\lambda$$

$$Z = (\frac{b^2}{a^2}N + h)\sin\varphi$$

where

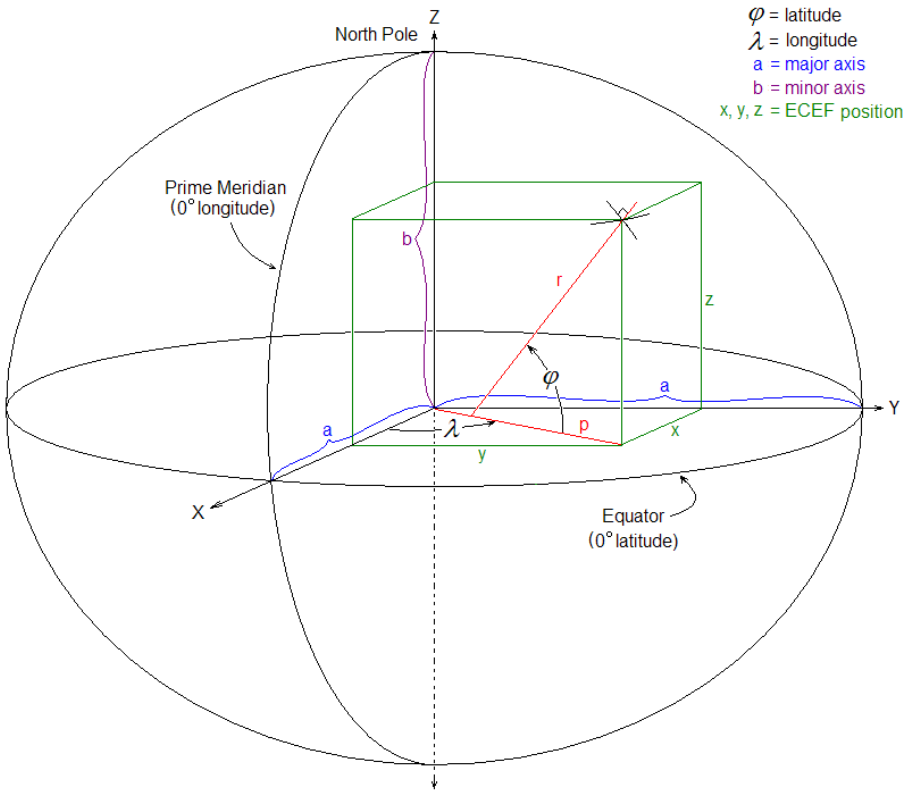

Figure A1 ECEF and Reference Ellipsoid

$$\varphi = latitude$$

$$\lambda = longitude$$

$$h = height\ above\ ellipsoid\ (meters)$$

$$N = Radius\ of\ Curvature\ (meters), defined\ as$$

$$= \frac{a}{\sqrt{1 - e^2 sin^2 \varphi}}$$

$$a = 6378137(meters)$$

$$b = a(1 - f)$$

f is the ellipsoidal flattening parameter defined by WGS84

$$f = \frac{1}{298.257223563}$$

e is the referred to as the eccentricity of the earth and can be thought of as a measure of how much the earth deviates from being a circle

$$e = \sqrt{\frac{a^2 - b^2}{b^2}}$$

Appendix B: MMR ARINC Data Logger

The ARINC data logger application is a command line application written in C++ with no user interface. Its function is to interface to Condor CEI-715 ARINC card. It reads, decodes, and logs the ARINC labels output by the MMR. The following figure is the primary method used to decode and log the ARINC labels.

```
void Logger::GetARINCLabels()
{
    short board = m_board;
    struct GNSS current;
    ARINC GNSS_word = 0;
    ARINC label;
    unsigned long lat = 0;
    unsigned long lon =0;
    int error_count = 1;
    int data_count = 0;

    //initialize GNSS struct to zero
    current = initGNSS();

    CString trace;

    trace =
"HEADER,UTC,Lat,Lon,Alt(m),HDOP,VDOP,HIL(m),VIL(m),SV_Visible,SV_Tracked,op_m
ode,ex_op_mode";
    //write the header to the log file
    LogFile.Trace(trace);

    while (TRUE)
    {
        if (check_for_quit() == TRUE)
            break;

        GNSS_word = get_ARINC_data(board,0);
```

```
if (GNSS_word == -1)
  {
      error_count ++;
                  printf("error count: %d      data count:
      %d\n",error_count, data_count);
  }
  else
  {
      data_count ++;

      SaveLabel(GNSS_word);

      //determine which label we have and decode it
      switch ((short)(label_mask & GNSS_word))
      {
          case label_110: //latitude measurement
              current.lat.raw = GNSS_word;
                                      current.lat.data =
                  (double)((data_mask & GNSS_word) >> 8) * 180 *
                  pow(2.0,-20);
              current.lat.ssm  = (short)((ssm_mask & GNSS_word) >> 29);
              current.lat.label = label_110;
              if ((sign_mask & GNSS_word) != 0)
              {
                  current.lat.data = current.lat.data -180;
              }
              current.valid = TRUE;
              break;
          case label_111: //longitude measurement
              current.lon.raw = GNSS_word;
              current.lon.data = (double)((data_mask & GNSS_word) >> 8)
              * 180 * pow(2.0,-20);
              current.lon.ssm  = (short)((ssm_mask & GNSS_word) >> 29);
              current.lon.label = label_111;
              if ((sign_mask & GNSS_word) != 0)
              {
                  current.lon.data = current.lon.data -180;
              }
              current.valid = TRUE;
              break;
          case label_120: //latitude fine measurement
              current.lat_fine.raw = GNSS_word;
              current.lat_fine.data =
             (double)((data_mask & GNSS_word)  >> 17) * 0.000172
             * pow(2.0,-11);
              current.lat_fine.ssm  =
            (short)((ssm_mask & GNSS_word)   >> 29);
              current.lat_fine.label = label_120;
              if ((sign_mask & GNSS_word) != 0)
              {
                  current.lat_fine.data =
                  current.lat_fine.data - 0.000172 * pow(2.0,-1);
              }
              current.valid = TRUE;
              break;
```

```
            case label_121: //longitude fine measurement
                 current.lon_fine.raw = GNSS_word;
                 current.lon_fine.data =
                (double)((data_mask & GNSS_word) >> 17) * 0.000172
                * pow(2.0,-11);
                 current.lon_fine.ssm  =
                (short)((ssm_mask & GNSS_word) >> 29);
                 current.lon_fine.label = label_120;

                 if ((sign_mask & GNSS_word) != 0)
                 {
                     current.lon_fine.data =
                      current.lon_fine.data - 0.000172 * pow(2.0,-1);
                 }
                 current.valid = TRUE;
                 break;
             case label_370: //WGS-84 height, feet
                 current.wgs_height.raw= GNSS_word;
                 current.wgs_height.data =
                (double)((data_mask & GNSS_word) >> 8) *  0.125;
                 current.wgs_height.ssm  =
               (short)((ssm_mask & GNSS_word) >> 29);
                 current.wgs_height.label = label_370;

                 if ((sign_mask & GNSS_word) != 0)
                 {
                     current.wgs_height.data =
                     current.wgs_height.data - 131072;
                 }
                 current.valid = TRUE;
                 break;
             case label_140: //UTC time fine measurement
                 current.UTC_time_fine.raw= GNSS_word;
                 current.UTC_time_fine.data =
                (double)((data_mask & GNSS_word) >> 8) *  pow(2.0,-20);
                 current.UTC_time_fine.ssm  =
                (short)((ssm_mask & GNSS_word) >> 29);
                 current.UTC_time_fine.label = label_140;

                 current.valid = TRUE;
                 break;
        case label_150: //UTC time
                 current.UTC_time.raw= GNSS_word;
                 current.UTC_time.hours   =
                 (short)((hour_mask & GNSS_word) >> 23);
                 current.UTC_time.minutes =
                 (short)((minute_mask & GNSS_word) >> 17);
                 current.UTC_time.seconds =
                 (short)((second_mask & GNSS_word) >> 11);
                 current.UTC_time.ssm  =
                 (short)((ssm_mask & GNSS_word) >> 29);
                 current.UTC_time.label = label_150;

                 current.valid = TRUE;
                 break;
```

```
case label_101: //HDOP
        current.HDOP.raw = GNSS_word;
        current.HDOP.data =
       (double)((data_mask & GNSS_word) >> 13)
        * 1024 * pow(2.0,-15);
        current.HDOP.ssm  =
       (short)((ssm_mask & GNSS_word) >> 29);
        current.HDOP.label = label_101;

        current.valid = TRUE;
        break;
case label_102: //VDOP
        current.VDOP.raw = GNSS_word;
        current.VDOP.data =
        (double)((data_mask & GNSS_word) >> 13)
        * 1024 * pow(2.0,-15);
        current.VDOP.ssm  =
       (short)((ssm_mask & GNSS_word) >> 29);
        current.VDOP.label = label_102;

        current.valid = TRUE;
        break;
case label_130: //Horizontal Integrity Limit is in Nautical Miles
        current.HIL.raw = GNSS_word;
        current.HIL.data =
        (double)((data_mask & GNSS_word) >> 11)
        * 16 * pow(2.0,-17);
        current.HIL.ssm = (short)((ssm_mask & GNSS_word) >> 29);
        current.HIL.label = label_130;

        current.valid = TRUE;
        break;
case label_133: //Vertical Integrity Limit is in Feet
        current.VIL.raw = GNSS_word;
        current.VIL.data =
        double)((data_mask & GNSS_word) >> 10)
        * 32768 * pow(2.0,-18);
        current.VIL.ssm = (short)((ssm_mask & GNSS_word) >> 29);
        current.VIL.label = label_133;

        current.valid = TRUE;
        break;
```

```
                case label_273:
                        current.status.raw = GNSS_word;
                        current.status.op_mode =
                        (short)((op_mode_mask & GNSS_word) >> 25) ;
                        current.status.ex_op_mode =
                        (short)((ex_op_mode_mask & GNSS_word) >> 23);
                        current.status.sv_tracked =
                        (short)((sv_tracked_mask & GNSS_word) >> 19) ;
                        current.status.sv_visible  =
                        (short)((sv_visible_mask & GNSS_word) >> 15) ;
                        current.status.ssm  =
                       (short)((ssm_mask & GNSS_word) >> 29);
                        current.status.label = label_273;
                    break;
                    default:
                        current.valid = 0;
                    break;
                }

                if (current.valid == TRUE)  //good data from MMR
                {

                    //build trace message

                      trace.Format("POSITION,%d:%d:%d,%lf,%lf,%lf,%lf,%lf,%lf,%lf
                      ,%d,%d,%d,%d"
                                    ,current.UTC_time.hours
                                    ,current.UTC_time.minutes
                                    ,current.UTC_time.seconds
                                    ,current.lat.data+current.lat_fine.data
                                    ,current.lon.data+current.lon_fine.data
                                    ,FeetToMeters(current.wgs_height.data)
                                    // convert feet to meters
                                    ,current.HDOP.data
                                    ,current.VDOP.data
                                    ,NMToMeters(current.HIL.data)
                                   //convert Nautical miles to meters
                                    ,FeetToMeters(current.VIL.data)
                                    //convert feet to meters
                                    ,current.status.sv_visible
                                    ,current.status.sv_tracked
                                    ,current.status.op_mode
                                    ,current.status.ex_op_mode);

                    //Log decoded message to trace file
                    LogFile.Trace(trace);

                }

            }
        }
    }
```

Appendix C: MMR ARINC Labels

GNLU Digital Label Outputs -GNSS Bus

Label 110: GNSS Present Position Latitude - [BNR]

BIT	FUNCTION	CODING	
1	Label 1st Digit	1 0	
2	Label 1st Digit	1	
3	Label 2nd Digit	1 0	
4	Label 2nd Digit	0	
5	Label 2nd Digit	1	
6	Label 3rd Digit	0 0	
7	Label 3rd Digit	0	
8	Label 3rd Digit	0	
9	Position Degrees	1 = 0.000171661	LSB = 180×2^{-20}
10	Position Degrees	1 = 0.000343323	
11	Position Degrees	1 = 0.000686646	
12	Position Degrees	1 = 0.001373291	
13	Position Degrees	1 = 0.002746582	
14	Position Degrees	1 = 0.005493164	
15	Position Degrees	1 = 0.010986328	
16	Position Degrees	1 = 0.021972656	
17	Position Degrees	1 = 0.043945313	
18	Position Degrees	1 = 0.087890625	
19	Position Degrees	1 = 0.17578125	
20	Position Degrees	1 = 0.3515625	Latitude = 0 to 360 Degrees
21	Position Degrees	1 = 0.703125	
22	Position Degrees	1 = 1.40625	
23	Position Degrees	1 = 2.8125	
24	Position Degrees	1 = 5.625	
25	Position Degrees	1 = 11.25	
26	Position Degrees	1 = 22.5	
27	Position Degrees	1 = 45	
28	Position Degrees	1 = 90	MSB = 180×2^{-1}
29	Sign Bit	0 = positive (North),	1 = -180 Degrees (South)
30	Sign Status Matrix (SSM)		
31	Sign Status Matrix (SSM)		
32	Parity (Odd)		

NOTES

[1] Update Rate = 1000 msec, Max Delay = 200 msec

SSM Code

BITS		MEANING
31	30	
0	0	Failure Warning
0	1	No Computed Data
1	0	Functional Test
1	1	Normal Operation

Figure C1 ARINC Label 110

GNLU Digital Label Outputs -GNSS Bus

Label 111: GNSS Present Position Long - [BNR]

BIT	FUNCTION	CODING	
1	Label 1st Digit	1 0	
2	Label 1st Digit	1	
3	Label 2nd Digit	1 0	
4	Label 2nd Digit	0	
5	Label 2nd Digit	1	
6	Label 2nd Digit	1 0	
7	Label 2nd Digit	0	
8	Label 2nd Digit	1	
9	Position Degrees	1 = 0.000171661	LSB = 180×2^{-20}
10	Position Degrees	1 = 0.000343323	
11	Position Degrees	1 = 0.000686646	
12	Position Degrees	1 = 0.001373291	
13	Position Degrees	1 = 0.002746582	
14	Position Degrees	1 = 0.005493164	
15	Position Degrees	1 = 0.010986328	
16	Position Degrees	1 = 0.021972656	
17	Position Degrees	1 = 0.043945313	
18	Position Degrees	1 = 0.087890625	
19	Position Degrees	1 = 0.17578125	
20	Position Degrees	1 = 0.3515625	Track Angle = 0 to 360 Degrees
21	Position Degrees	1 = 0.703125	
22	Position Degrees	1 = 1.40625	
23	Position Degrees	1 = 2.8125	
24	Position Degrees	1 = 5.625	
25	Position Degrees	1 = 11.25	
26	Position Degrees	1 = 22.5	
27	Position Degrees	1 = 45	
28	Position Degrees	1 = 90	MSB = 180×2^{-1}
29	Sign Bit	0 = positive (East of North),	1 = -180 Degrees (West of North)
30	Sign Status Matrix (SSM)		
31	Sign Status Matrix (SSM)		
32	Parity (Odd)		

NOTES

[1] Update Rate = 1000 msec, Max Delay = 200 msec

SSM Code

BITS		MEANING
31	30	
0	0	Failure Warning
0	1	No Computed Data
1	0	Functional Test
1	1	Normal Operation

Figure C2 ARINC Label 111

GNLU Digital Label Outputs -GNSS Bus

Label 120: Latitude Fine - [BNR]

BIT	FUNCTION	CODING			
1	Label 1st Digit	1 0			
2	Label 1st Digit	1			
3	Label 2nd Digit	2 0			
4	Label 2nd Digit	1			
5	Label 2nd Digit	0			
6	Label 3rd Digit	0 0			
7	Label 3rd Digit	0			
8	Label 3rd Digit	0			
9	SDI Code	0 All	1 Left	0 Right	1 3rd
10	SDI Code	0 Call	0 Unit	1 Unit	1 Unit
11	PAD				
12	PAD				
13	PAD				
14	PAD				
15	PAD				
16	PAD				
17	PAD				
18	Position Degrees	1 = 8.39844E-08 LSB = 0.000172×2^{-11}			
19	Position Degrees	1 = 1.67969E-07			
20	Position Degrees	1 = 3.35938E-07			
21	Position Degrees	1 = 6.71875E-07			
22	Position Degrees	1 = 1.34375E-06 Latitude Fine Degrees = 0 to 0.00172 Degrees			
23	Position Degrees	1 = 2.6875E-06			
24	Position Degrees	1 = 0.000005375			
25	Position Degrees	1 = 0.00001075			
26	Position Degrees	1 = 0.0000215			
27	Position Degrees	1 = 0.000043			
28	Position Degrees	1 = 0.000086 MSB = 0.000172×2^{-1}			
29	Sign BIT	0 = Positive (North) 1 = -0.000172×2^{-1} (South)			
30	Sign Status Matrix (SSM)				
31	Sign Status Matrix (SSM)				
32	Parity (Odd)				

NOTES

[1] Fine Data Words contain truncated position of the original data word (Label 110). The two labels are concatenated in the receiver.

[2] Update Rate = 1000 msec, Max Delay = 200 msec

SSM Code

BITS		MEANING
31	30	
0	0	Failure Warning
0	1	No Computed Data
1	0	Functional Test
1	1	Normal Operation

Figure C3 ARINC Label 120

GNLU Digital Label Outputs -GNSS Bus

Label 121: Long Fine - [BNR]

BIT	FUNCTION	CODING			
1	Label 1st Digit	1	0		
2	Label 1st Digit		1		
3	Label 2nd Digit	2	0		
4	Label 2nd Digit		1		
5	Label 2nd Digit		0		
3	Label 2nd Digit	1	0		
4	Label 2nd Digit		0		
5	Label 2nd Digit		1		
9	SDI Code	0 All	1 Left	0 Right	1 3rd
10	SDI Code	0 Call	0 Unit	1 Unit	1 Unit
11	PAD				
12	PAD				
13	PAD				
14	PAD				
15	PAD				
16	PAD				
17	PAD				
18	Position Degrees	1 = 8.39844E-08			LSB = 0.000172×2^{-11}
19	Position Degrees	1 = 1.67969E-07			
20	Position Degrees	1 = 3.35938E-07			
21	Position Degrees	1 = 6.71875E-07			
22	Position Degrees	1 = 1.34375E-06			Longitude Degrees = 0 to 0.00172 Degrees
23	Position Degrees	1 = 2.6875E-06			
24	Position Degrees	1 = 0.000005375			
25	Position Degrees	1 = 0.00001075			
26	Position Degrees	1 = 0.0000215			
27	Position Degrees	1 = 0.000043			
28	Position Degrees	1 = 0.000086			MSB = 0.000172×2^{-1}
29	Sign BIT	0 = positive (East of North), 1 = -0.000172×2^{-1} Degrees (West of North)			
30	Sign Status Matrix (SSM)				
31	Sign Status Matrix (SSM)				
32	Parity (Odd)				

NOTES

[1] Fine Data Words contain truncated position of the original data word (Label 111). The two labels are concatenated in the receiver. Update Rate = 1000 msec, Max Delay = 200 msec

SSM Code

BITS		MEANING
31	30	
0	0	Failure Warning
0	1	No Computed Data
1	0	Functional Test
1	1	Normal Operation

Figure C4 ARINC Label 121

GNLU Digital Label Outputs -GNSS Bus

Label 140: UTC, Fine (GPS) [BNR]

BIT	FUNCTION	CODING	
1	Label 1st Digit	1 0	
2	Label 1st Digit	1	
3	Label 2nd Digit	4 1	
4	Label 2nd Digit	0	
5	Label 2nd Digit	0	
6	Label 3rd Digit	0 0	
7	Label 3rd Digit	0	
8	Label 3rd Digit	0	
9	1= 9.53674E-07		1x2E-20 Seconds (LSB)
10	1= 1.90735E-06		
11	1= 3.8147E-06		
12	1= 7.62939E-06		
13	1= 1.52588E-05		
14	1= 3.05176E-05		
15	1= 6.10352E-05		
16	1= 0.00012207		
17	1= 0.000244141		
18	1= 0.000488281		Binary Sequence UTC, Fine (Seconds)
19	1= 0.000976563		(Two's Complement)
20	1= 0.001953125		
21	1= 0.00390625		
22	1= 0.0078125		
23	1= 0.015625		
24	1= 0.03125		
25	1= 0.0625		
26	1= 0.125		
27	1= 0.25		
28	1= 0.5		1x2E-1 Seconds (MSB)
29	Sign BIT		Set to 0 (Always Positive)
30	SSM [3]		
31	SSM [3]		
32	Parity (Odd)		

NOTES
[1] Fine Data Words contain truncated position of the original data word (Label 150). The two labels are concatenated in the receiver.
[2] Update Rate = 1000 msec, Max Delay = 200 msec
[3] Sign Status Matrix (SSM)

BITS		MEANING
31	30	
0	0	Failure Warning
0	1	No Computed Data
1	0	Functional Test
1	1	Normal Operation

Figure C5 ARINC Label 140

GNLU Digital Label Outputs -GNSS Bus

Label 150: UTC or GPS Time - [BNR/DIS]

BIT	FUNCTION	CODING			
1	Label 1st Digit	1	0		
2	Label 1st Digit		1		
3	Label 2nd Digit	5	1		
4	Label 2nd Digit		0		
5	Label 2nd Digit		1		
6	Label 3rd Digit	0	0		
7	Label 3rd Digit		0		
8	Label 3rd Digit		0		
9	SDI Code	0 All	1 Left	0 Right	1 3rd
10	SDI Code	0 Call	0 Unit	1 Unit	1 Unit
11	UTC Time Source*	1 = GNSS System Clock	0 = External Clock		
12	Seconds	1= 1			
13	Seconds	1= 2			
14	Seconds	1= 4	Seconds		
15	Seconds	1= 8			
16	Seconds	1= 16			
17	Seconds	1= 32			
18	Minutes	1= 1			
19	Minutes	1= 2			
20	Minutes	1= 4	Minutes		
21	Minutes	1= 8			
22	Minutes	1= 16			
23	Minutes	1= 32			
24	Hours	1= 1			
25	Hours	1= 2			
26	Hours	1= 4	Hours		
27	Hours	1= 8			
28	Hours	1= 16			
29	0 = Always Positive				
30	SSM [3]				
31	SSM [3]				
32	Parity (Odd)				

NOTES
[1]*Bit 11 is set to "1" when the GNSS system clock is being used as the source of time. Otherwise, set to "0" when time is based on internal or external clock.
[4] Update Rate = 1000 msec, Max Delay = 200 msec
[5] Sign Status Matrix (SSM)

BITS		MEANING
31	30	
0	0	Failure Warning
0	1	No Computed Data
1	0	Functional Test
1	1	Normal Operation

Figure C6 ARINC Label 150

GNLU Digital Label Outputs -GNSS Bus

Label 273: GNR Sensor Status [DISC]

BIT	FUNCTION	CODING
1	Label 1st Digit	2 1
2	Label 1st Digit	0
3	Label 2nd Digit	7 1
4	Label 2nd Digit	1
5	Label 2nd Digit	1
6	Label 3rd Digit	3 0
7	Label 3rd Digit	1
8	Label 3rd Digit	1
9	SDI Code	0 All 1 Left 0 Right 1 3rd
10	SDI Code	0 Call 0 Unit 1 Unit 1 Unit
11	MSB of Satellites Visible	1= >15 Visible 0 = < 15 Visible
12	DADC/FMS Status	1 = Not Present 0 = Present
13	DADC/FMS Source	1 = Cross-side 0 = On-side
14	IRS/FMS Status	1 = Not Present 0 = Present
15	IRS/FMS Source (1)	1 = Cross-side 0 = On-side
16	Number of Sat Visible	LSB (2)
17	Number of Sat Visible	
18	Number of Sat Visible	
19	Number of Sat Visible	MSB
20	Number of Sat Tracked	LSB (2)
21	Number of Sat Tracked	Number of Satellites Tracked (0 to 15)
22	Number of Sat Tracked	
23	Number of Sat Tracked	MSB
24	Reserved For Expanded Sensor Operation Mode	
25	Reserved For Expanded Sensor Operation Mode	
26	GPS Sensor Operational Mode	
27	GPS Sensor Operational Mode	Refer to Table 19
28	GPS Sensor Operational Mode	
29	MSB of Satellites Tracked	1 = > 15 Track (rsvd) 0 = 15 or less Tracked
30	SSM [3]	
31	SSM [3]	
32	Parity (Odd)	

NOTES
[1] Secondary will indicate that SysP LCL TX is inactive and that at least one IRS RX bus is active. If the SysP LCL TX bus is active the source shall be Primary.
[2] Satellites Tracked is number of satellites tracked and used for PVT solution.
[3] Sign Status Matrix (SSM), See Table GNSS273-2
[4] Update Rate = 1000 msec

Table GNSS273-2		
BITS		MEANING
31	30	
0	0	Normal Operation
0	1	Not Used (No Computed Data)
1	0	Functional Test
1	1	Not Used

Table 19 - GNSS Bus Label 273 GPS Sensor Op Mode			
28	27	26	MODE
0	0	0	Self Test
0	0	1	Initialization
0	1	0	Acquisition
0	1	1	Navigation
1	0	0	Altitude Aiding
1	0	1	Differential
1	1	0	Direction/Speed Aiding
1	1	1	Fault

Figure C7 ARINC Label 273

GNLU Digital Label Outputs -GNSS Bus

Label 370: GNSS Height (WGS-84) [BNR]

BIT	FUNCTION	CODING	
1	Label 1st Digit	3	1
2	Label 1st Digit		1
3	Label 2nd Digit	7	1
4	Label 2nd Digit		1
5	Label 2nd Digit		1
6	Label 3rd Digit	0	0
7	Label 3rd Digit		0
8	Label 3rd Digit		0
9	0.125		
10	0.25		
11	0.5		
12	1		
13	2		
14	4		
15	8		
16	16		
17	32		
18	64		
19	128	(FEET)	
20	256		
21	512		
22	1024		
23	2048		
24	4096		
25	8192		
26	16384		
27	32768		
28	65536		
29	0 = PLUS, 1 = -131,072 FEET		
30	SSM Code [1]		
31	SSM Code [1]		
32	Parity (odd)		

NOTE: This label is the same as label 076, except the height above WGS-84 is given

[1] Status Sign Matrix (SSM)

BITS		MEANING
31	30	
0	0	Failure Warning
0	1	No Computed Data
1	0	Functional Test
1	1	Normal Operation

Figure C8 ARINC Label 370

Appendix D: Course Deviation Indicator Wiring Diagram

The following wiring diagram was acquired from a Cessna Service and Parts Manual and was used to instrument the Course Deviation Indicator to read both the Glideslope flag and Vertical Pointer voltages to detect availability.

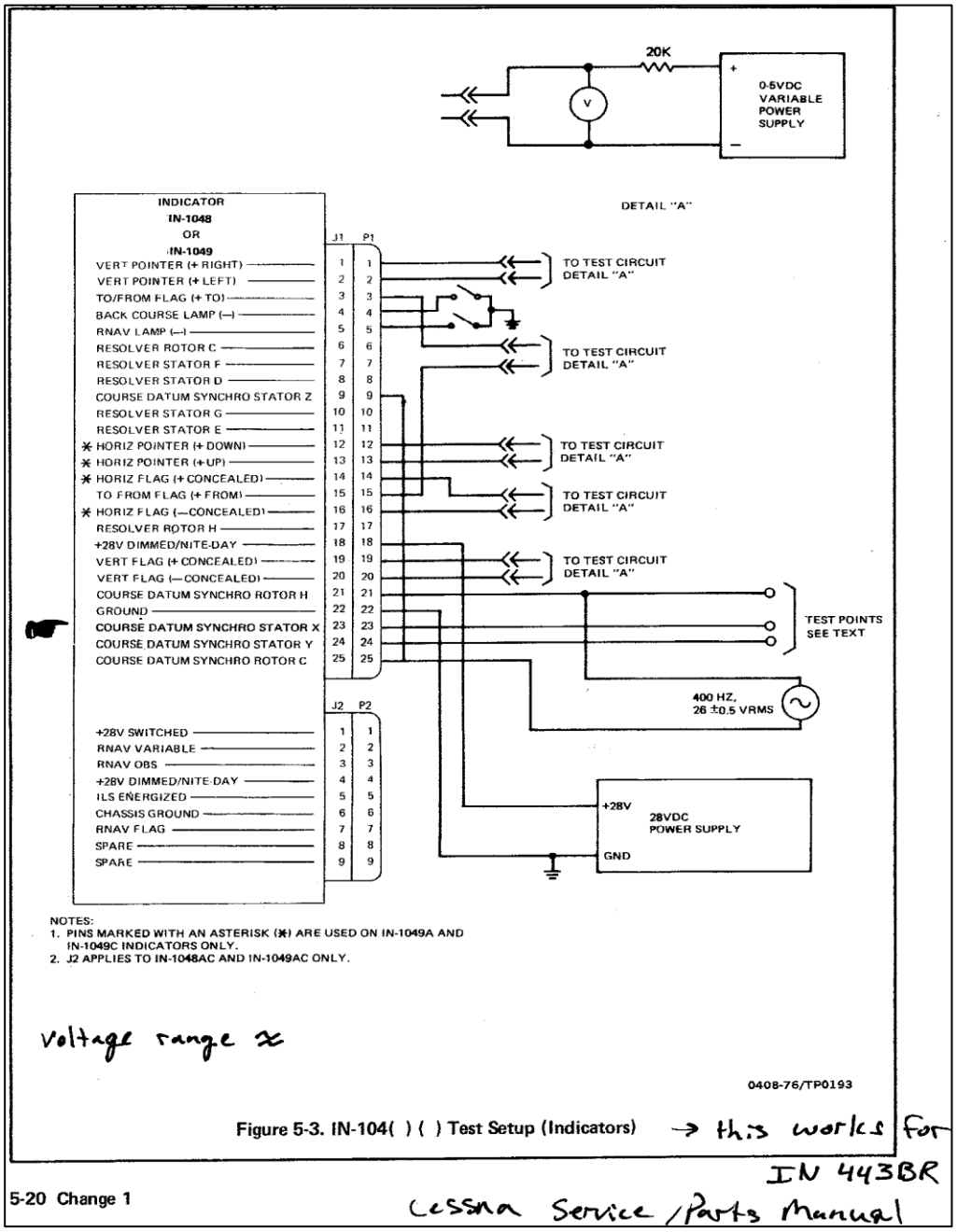

Figure D1 Course Deviation Indicator Wiring Diagram

Appendix E: MMR ARINC Log File Reader

The figure below is the Graphical User Interface (GUI) of the custom ARINC log reader application written to read and process the log files generated by the MMR ARINC data logger described in Appendix B.

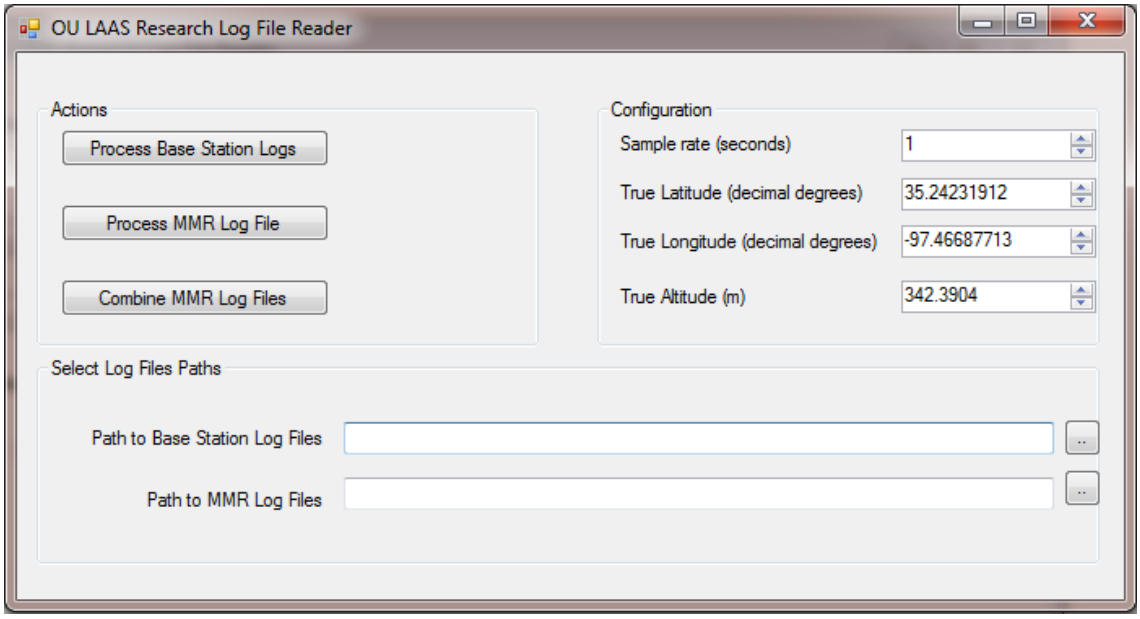

Figure E1 MMR ARINC Log File Reader GUI

The following C# code is the primary method used to parse, down-sample, and process the log files generated by the MMR ARINC data logger.

```
public void parseMMRLogFile(int sampleRateInSeconds,
                                    ECEF referencePosition,
                                    string inputfilePath)
        {
            #region variable initialization

            string outputFileHeader = "MMR Time
(UTC),LocalTime(24hr),Lat,Lon,Alt(m),HDOP,VDOP,HIL(m),VIL(m),SV_Visible,SV_Tr
acked,op_mode,ex_op_mode,DiffMode,Alt_Averaged(m),True_Altitude,VE(m),
HE(m)";
            string delimStr = ",";
            char[] delimiter = delimStr.ToCharArray();

            int maxCoastSamples = 90;
            int coastCounter = 0;
            int HELimit = 1;

            int counter = 0;
            string currentLine = string.Empty;
            int lastSecond = -1;
            double accumulatedAltitude = 0;
            int accumulatedSamples = 0;
            double previousHorizontalError = 0;
            double accumulatedLatitude = 0;
            double accumulatedLongitude = 0;

            #endregion

            string outputFilePath = GetOutputFilePath(inputfilePath,
            sampleRateInSeconds);

            //read MMR Logfile line by line and process
            using (System.IO.StreamReader inputFile =
                  new System.IO.StreamReader(inputfilePath))
            {
                using (System.IO.StreamWriter outputFile =
                        new System.IO.StreamWriter(outputFilePath))
                {
                    outputFile.WriteLine(outputFileHeader);

                    while ((currentLine = inputFile.ReadLine()) != null)
                    {
                        List<string> lineParts =
                        currentLine.Split(',').ToList<string>();

                        if (lineParts[0].Equals("HEADER",
                         StringComparison.InvariantCultureIgnoreCase))
                        {
                            continue;
                        }
```

```
//fetch the part of the line that contains the MMR time
      string mmrTime = lineParts[1];
      //seperate the mmrTime into its parts
      List<string> mmrTimeParts =
      mmrTime.Split(':').ToList<string>();

      //keep a running average of Altitude
      double altitude = Convert.ToDouble(lineParts[4]);
      accumulatedAltitude = accumulatedAltitude + altitude;
      accumulatedSamples++;

      double latitudeCurrent =
      Convert.ToDouble(lineParts[2]);
      double longitudeCurrent =
      Convert.ToDouble(lineParts[3]);

      accumulatedLatitude =
      accumulatedLatitude + latitudeCurrent;
      accumulatedLongitude =
      accumulatedLongitude + longitudeCurrent;

      //parse out the seconds so we can count sample rate.
      int second = Convert.ToInt32(mmrTimeParts[2]);

      //only look at a record when the second changes
      if (second != lastSecond)
      {
          lastSecond = second;

          //only write data out
         //according to our sample rate
          if ((counter == 0))
          {
              string mmrDate =
             GetMMR_UTCDateFromLocalDate(localDate,
             localTime, mmrTime);

             int diffMode = GetDiffModeStatusFromMMR(
             lineParts[12]);

              double averagelatitude =
              accumulatedLatitude / accumulatedSamples;
              double averagelongitude =
              accumulatedLongitude / accumulatedSamples;
              double averageAltitude =
              accumulatedAltitude / accumulatedSamples;
              double averageVerticalError =
              averageAltitude - referencePosition.Altitude;
```

```
double averageHorizontalError =
GetHorizontalErrorInMeters(averagelatitude,
averagelongitude, altitude, referencePosition);
double horizontalErrorCurrent =
GetHorizontalErrorInMeters(latitudeCurrent,
longitudeCurrent,altitude,referencePosition);

                if (averageHorizontalError > HELimit)
                {
                    if (coastCounter < maxCoastSamples)
                    {
                        averageHorizontalError =
                        previousHorizontalError;
                        coastCounter++;
                    }
                    else
                    {
                        coastCounter = 0;
                        previousHorizontalError =
                        averageHorizontalError;
                    }
                }
                else
                {
                    coastCounter = 0;
                }

                previousHorizontalError =
                    averageHorizontalError;

                double horizontalErrorToLog = 0;

                if (horizontalErrorCurrent <
                          averageHorizontalError)
                {
                    horizontalErrorToLog =
                    horizontalErrorCurrent;
                }
                else
                {
                    horizontalErrorToLog =
                    averageHorizontalError;
                }

                //fetch the remaining data that we will add
                string allDataToTheRigtOfMMRTime =
                    currentLine.Split(delimiter, 3)[2];
```

```
                    //format the Altitude with 6 decimal points
                    string output =
                    string.Format("{0} {1},{2} {3},{4},{5},{6},{7},{8},{9}",
                                mmrDate,
                                mmrTime,
                                localDate,
                                localTime,
                                allDataToTheRigtOfMMRTime,
                                diffMode.ToString(),
                                averageAltitude.ToString("F6"),
                                referencePosition.Altitude.ToString("F6"),
                                averageVerticalError.ToString("F6"),
                                horizontalErrorToLog.ToString("F6"));
                                outputFile.WriteLine(output);
                            }

                            if (counter == (sampleRateInSeconds - 1))
                            {
                                counter = 0;
                                continue;
                            }

                            counter++;
                        }

                    }
                }
            }

        }
```

Appendix F: CEI-715

The Condor Engineering® CEI-715 is a PC Card (PCMCIA) interface that provides 12 fully independent ARINC 429 channels. This PC Card was used to read all ARINC messages output by the MMR.



Figure F1 Condor Engineering CEI-715