UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

A Real Time Ground Based Augmentation System Implemented with LabVIEW Software

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

David Milligan II
Norman, Oklahoma
2022

A Real Time Ground Based Augmentation System
Implemented with LabVIEW Software

A THESIS APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Yan Zhang, Chair

Dr. Chad Davis

Dr. Paul Moses

Acknowledgments

I would like to thank my committee members for their support throughout my time as a graduate student. Dr. Zhang has been a driving force keeping me moving in the right direction. Thank you Dr. Davis for years of mentorship and support, I would not be where I am without you. Thanks to Dr. Dyer for your mentorship during my thesis process and professionally. I would like to thank Jaxon Taylor, my fellow graduate student, for working through all of the hardware setup with me and support throughout the process. Thanks to Melany Arnold for her help and support everyday throughout my time in graduate school. Thanks to my father David Milligan who inspires me to be a better engineer every day. I would like to thank Jacob Henderson and Adam Click for their help along the way. Thank you to everyone else who has helped who is not listed here and every friend and family member that has provided me encouragement.

Contents

List Of Tables

List Of Figures

Abstract

This thesis develops and implements a real time Ground Based Augmentation System (GBAS) in the LabVIEW environment incorporating the Local Monitor (LM) from the Closed-Loop GBAS (CL-GBAS) concept. GBAS, also referred to as the Local Area Augmentation System (LAAS), is a flight landing system that provides landing guidance information and corrections to global positioning system (GPS) for aircraft. CL-GBAS is an implementation of LAAS that was prototyped in 2007 and uses the LM and Far Field Monitor (FFM) integrity monitoring systems to assess the accuracy of the GPS corrections generated in the system and to assess the validity of the data broadcast.

The software architecture leverages the robust timing and pipelining features of LabVIEW to perform data acquisition and synchronization, perform the LAAS algorithm to generate Type 1 corrections, perform the CL-GBAS algorithm to evaluate the accuracy of the GPS corrections, and perform communication with the VHF Data Broadcast (VDB) transmitter. The LAAS ground facility (LGF) was successfully constructed and integrated with GPS receivers and a VDB transmitter and the accuracy of the GBAS corrections were determined to perform at the highest GBAS Service Level (GSL F).

Chapter 1

Introduction and Background

This study focuses on a real time implementation of the Ground Based Augmentation System (GBAS) that uses a Local Monitor (LM) as defined in Closed-Loop GBAS (CL-GBAS) to generate and assess the accuracy of measurements that are generated from the system. This work presents the hardware used to develop and test the GBAS and the real time software architecture that was developed in LabVIEW, including the GPS receiver data acquisition and synchronization subsystem, the VHF Data Broadcast (VDB) interface, and the CL-GBAS algorithm implementation.

The Local Area Augmentation System (LAAS) is a flight landing system that provides landing guidance information and corrections to global positioning system (GPS) for aircraft. LAAS generates Type 1 correction messages using differential GPS (DGPS) techniques with GPS measurements that are taken from multiple local reference stations that are used by a navigator to correct a GPS position solution. LAAS Type 2 messages contain information that describe the coordinates of the LGF and data used to compute a tropospheric correction [1]. LAAS Type 3 messages are null messages used in VDB authentication protocols [1]. LAAS Type 4 messages contain the final approach segment (FAS) data related to the path that an aircraft would follow during landing. LAAS is referred to as GBAS because all of the reference stations, data processing, and data transmission are performed on the ground. CL-GBAS is an implementation of LAAS that uses integrity monitoring systems to assess the accuracy of the GPS corrections generated in the system and to assess the validity of the data broadcast from the system. The integrity network of a CL-GBAS is composed of a LM that applies corrections to a GPS receiver at a known location to assess the accuracy of the corrections and a Far Field Monitor (FFM) that receives and assesses the quality of broadcast data from a point inside the airfield [2]. CL-GBAS was developed as a prototype in 2007 using simulations but has not been implemented in real time systems [3].

1.1 Background

Modern aviation relies upon flight navigation and landing systems. The Instrument Landing System (ILS) is a ground based landing system that uses two pairs of VHF carrier signals to indicate a landing glide slope and position relative to the runway [4]. The VHF Omni-directional Range (VOR) navigation system is composed of beacons that are used to provide an aircraft with azimuth and can be used for high and low altitude routes and for airport approaches [5]. The Global Positioning System (GPS) uses a constellation of satellites as navigation location reference points [6]. Each GPS satellite transmits data related to the satellite location and data that can be used to calculate the distance to the satellite. Differential GPS (DGPS) systems are systems that use information collected from other receivers and use that information to improve the position solutions possible via traditional GPS. The Local Area Augmentation System (LAAS) is a ground based version of DGPS that provides correction data to aircraft during their airport approaches based on information collected that is specific to that airport [7].

1.1.1 Global Positioning System (GPS)

GPS is a navigation system that was developed by the United States government to fill a need for precise timing and positioning for military and security purposes. Since the first GPS satellite launch on February 22, 1978 the GPS constellation has grown and retired many satellites retaining 32 active satellites in medium earth orbit. In 1996, the United States government allowed civilians access to the data from the GPS constellation after applying selective availability (SA) to the system. SA applied an intentional error to the satellite clock and the satellite ephemeris that was greater than five times the error inherent in the system [8]. On May 02, 2000 SA was turned off by executive order allowing for high accuracy GPS solutions to be used in a civilian context. The precise positioning service (PPS) is a subset of GPS data that is available to the military that provides high accuracy. The standard positioning service (SPS) is the subset of GPS data that is available to civilians and is less accurate than the PPS [9] .

GPS is an evolution of the trilateration approach to navigation, where at least three radio sources generate signals that can be used to infer distance. Measuring the distances from three known locations allows the navigator to determine its location on a plane. In GPS, a similar method involving at least four satellites is used. Satellites are referred to as space vehicles (SVs) in the GPS context. The

GPS signal encodes information about the time of signal transmission and the location of the SV. The difference between the time of transmission and the time of arrival (TOA) can be used to calculate the distance between the SV and the navigator. This distance is referred to as the pseudorange because it is an estimate or range rather than a true measurement. The 4th SV is used to address the receiver clock bias caused by the synchronization difference between the high precision nuclear clocks in the SV and the relatively low precision clocks in the navigator receiver.

The GPS architecture is divided into the space, control, and user segments. The space segment is comprised of up to 32 SVs traveling along a radius of 26,560 km over approximately a 12 hour period. The SVs are organized using a two letter code that defines their orbital plane and their slot within that plane. The orbital planes, labeled A through F, have four primary slots and up to two spare slots. When the orbital planes have all of their primary slots filled, a minimum of four SVs are available almost anywhere on earth under good atmospheric conditions [9]. The control segment, also known as the Master Control Station (MCS), is housed in Schriever Air Force Base in Colorado. The control segment is responsible for monitoring and maintaining the health, orbit, and timing of the SVs. The MCS may update SV navigation messages or maneuver SVs to maintain orbits or correct for SV failures. The user segment is comprised of all GPS receivers that use SPS or PPS signals to navigate.

GPS receivers collect data in the L-band on the Link 1 (L1) and Link 2 (L2) frequencies. The center frequency for L1 is 1575.42 MHz and the center frequency for L2 is 1227.60 MHz. L1 contains both civilian unencrypted signals and government encrypted signals, while L2 contains only government encrypted signals. The GPS signal generated by each SV is composed of a carrier sinusoid, a ranging code, and navigation data. The carrier sinusoid operates at L1 or L2 frequencies. The ranging code is a psuedo-random number (PRN) that contains information that can be used to determine the range from the SV, uniquely identify the SV and receive transmissions from multiple SVs without interference on the same frequency [9]. The length and rate of the ranging code is determined by whether it is a SPS coarse/acquisition code (C/A-code) or a PPS precision code (P(Y)-code). C/A-codes are smaller and slower than the P(Y)-codes. The navigation data is a relatively slowly broadcast signal that contains satellite health information, ephemeris, almanac, and clock bias. The standard GPS accuracy is approximately 15 meters in the horizontal direction [10].

There are several sources of error in basic GPS solutions. Clock error is contributed by both the atomic clock present in the SV that functions as a time source for the GPS transmitter and the relatively low performance clock present on the GPS receiver [11]. Receiver clock error is treated as a fourth unknown in a standard GPS solution. This mitigates much of the error contribution from the receiver clock error down to approximately 2.1 meters [12]. The satellite ephemeris is a set of data that is used to calculate the location of an SV. Ephemeris error is present in a GPS solution when the parameters present in the ephemeris do not accurately model the position of an SV. Multipath error is present in a solution when the transmitted GPS signal is reflected off of objects in the environment and the reflections combine with the direct GPS signal [13]. A reflected signal may have a slightly different amplitude and phase than direct

GPS signal causing the GPS receiver to take a less accurate measurement.

Atmospheric error contributions make up a large portion of GPS error and can be divided into that which is caused by the troposphere and that which is caused by the ionosphere. Atmospheric error contributions become greater as the elevation of an SV over the horizon becomes lower. The troposphere is the layer of the atmosphere that extends from the ground to approximately 50 kilometers above ground. The troposphere contributes a delay to the code and carrier components of the GPS signal but makes up a relatively low proportion of the atmospheric error. The ionosphere is the layer of the atmosphere that extends from 50 kilometers to approximately 1000 kilometers above ground level. The thickness of the ionosphere varies unpredictably and diurnally. The ionosphere refracts the GPS signal causing the path of the GPS signal to increase in distance [14]. This causes an overall delay in the GPS measurement. The ionosphere adds further error because the carrier becomes accelerated while the code becomes delayed while traveling through this region.

1.1.2 Differential GPS (DGPS)

DGPS techniques are those that use GPS measurements from surveyed locations to generate corrections that are used to improve the accuracy of GPS positioning. Ground based augmentation systems (GBAS) are systems that use measurements taken from the ground and use transmission stations located on the ground. Wide Area Augmentation System (WAAS) is a DGPS system that uses GPS stations across the United States to correct the GPS SPS. WAAS is considered a satellite based augmentation system (SBAS) because it transmits

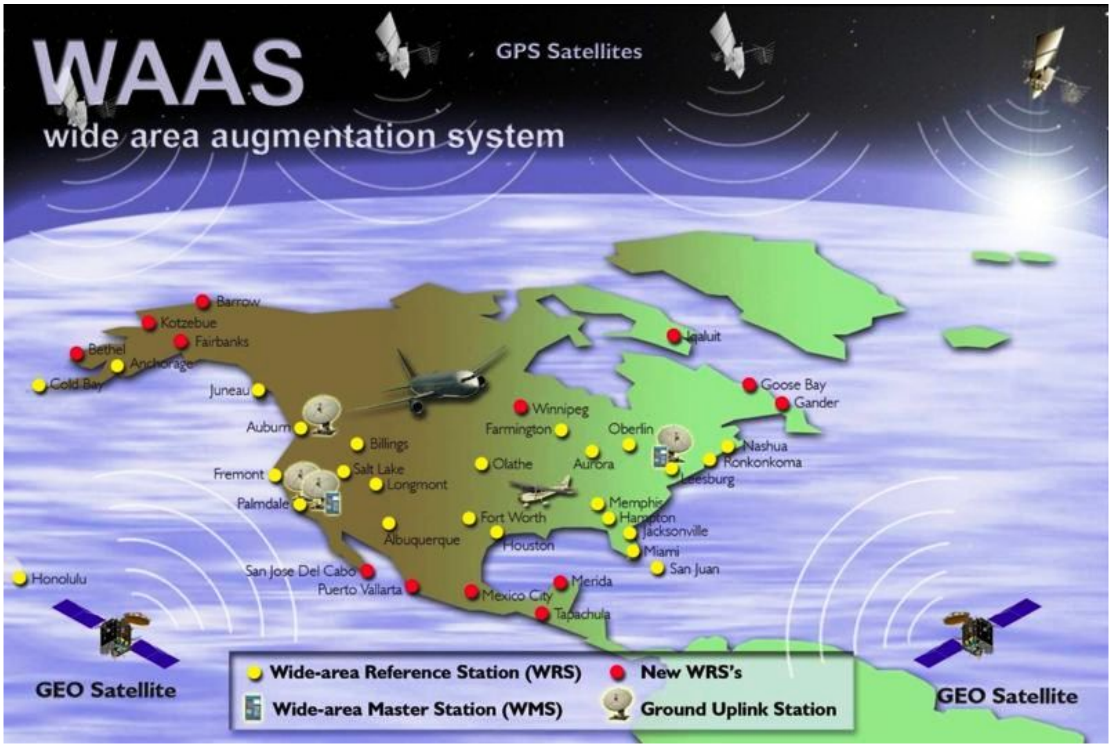


Figure 1.1. ***WAAS Architecture****: WAAS uses ground based measurements and a space based transmission system. WAAS is sometimes referred to as SBAS [16].*

corrections from communication satellites. WAAS communication satellites are geostationary and provide additional ranging sources for WAAS solutions. WAAS can provide corrections to the entirety of North America [15]. Figure 1.1 depicts the WAAS system architecture [16].

LAAS is a GBAS that uses a minimum of three local reference stations (RS) to generate corrections over a small area (20 to 30 miles) [17]. LAAS is intended to be used as an aviation navigation system and replacement to the instrument landing system (ILS) [3]. The LAAS corrections are generated from pseudorange and phase measurements taken by the RS. The LAAS ground facility (LGF) takes the measurements from the RS and generates Type 1 messages that contain the corrections and other pertinent information that can be used by a

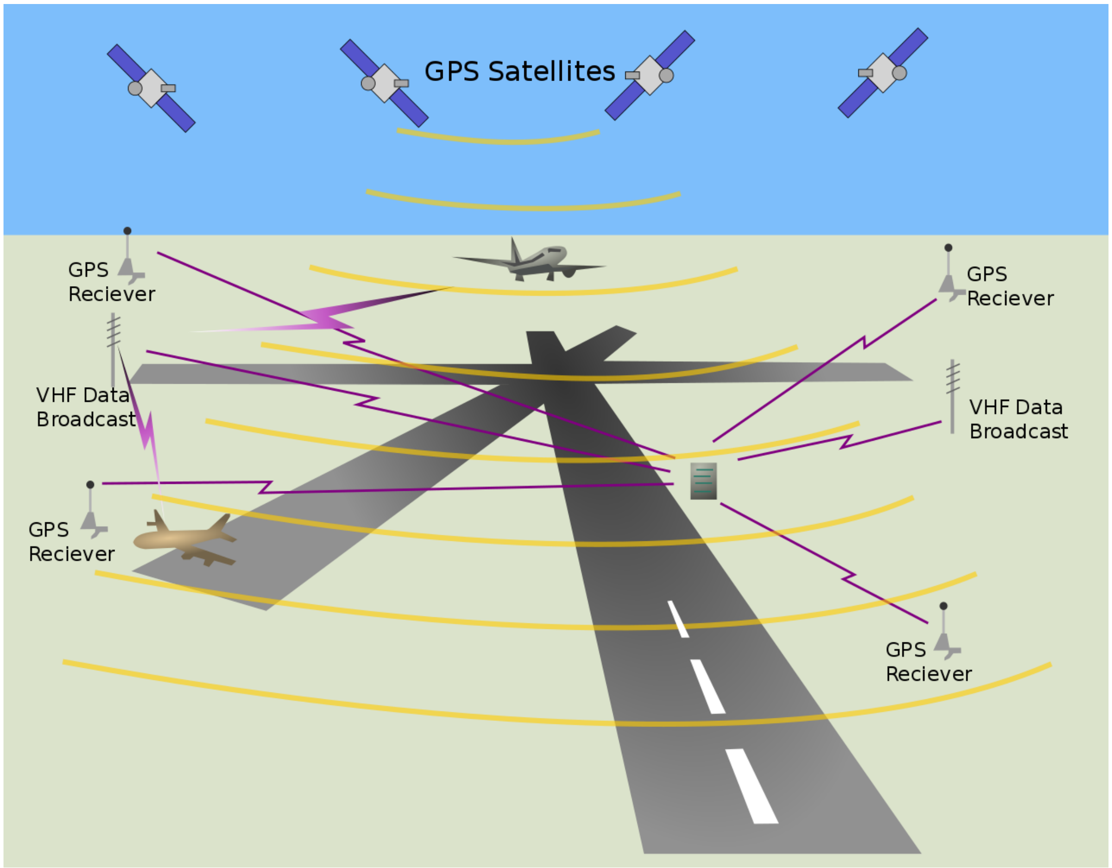


Figure 1.2. ***LAAS Architecture***: *LAAS uses ground based measurements and a ground based transmission system. LAAS is sometimes referred to as LAAS [19].*

navigator to correct their position. Corrections are broadcast to the service area using a VHF data broadcast system (VDB) [18]. Figure 1.2 depicts the LAAS system architecture [19].

Chapter 2

Architecture of the Hardware Assembled for OU LAAS Implementation

The LAAS implemented at OU is composed of four reference stations (RS) that send raw pseudorange measurement data to a base station where the LAAS algorithms are performed to generate corrections. GPS corrections are formatted into Type 1 messages. Type 2 and Type 4 LAAS messages contain information related to the LGF and final approach segment (FAS) respectively [1]. LAAS messages are sent to a VHF Data Broadcast (VDB) transmitter and broadcast to the airfield. Figure 2.1 is a high level diagram that depicts the flow of data through the hardware used in the LAAS system.

2.1 GPS Receivers

The GPS receivers used for the LAAS system must be capable of providing the information necessary to calculate the position of the SVs, the raw range from the SVs (measured in time), the carrier phase of the SVs, the receive time of the data, and the health status of each SV. Ephemeris data is used to calculate

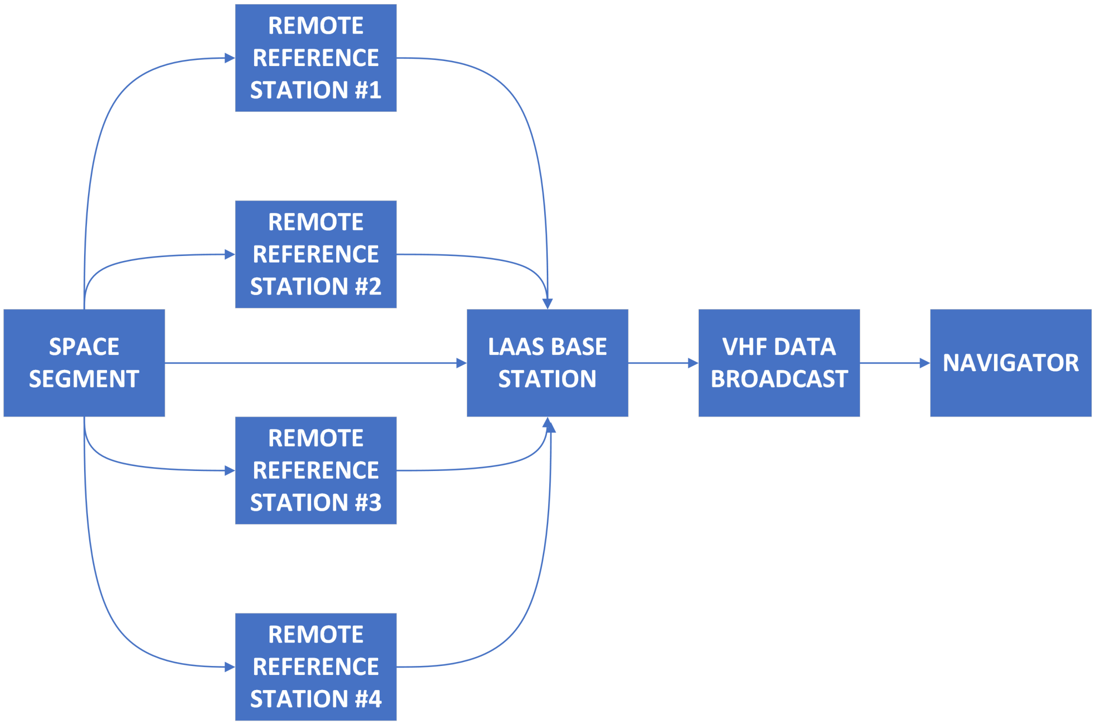

REMOTE REFERENCE STATION #1
REMOTE REFERENCE STATION #2
SPACE SEGMENT
LAAS BASE STATION
VHF DATA BROADCAST
NAVIGATOR
REMOTE REFERENCE STATION #3
REMOTE REFERENCE STATION #4

Figure 2.1. ***Hardware Data Flow Diagram**: Pseudorange data is captured from four RSs and one base station. The captured pseudorange data from each RS is sent to the base station for use in the LAAS algorithm. The VDB is used to transmit LAAS messages that are generated from the LAAS algorithm to the aircraft navigators.*

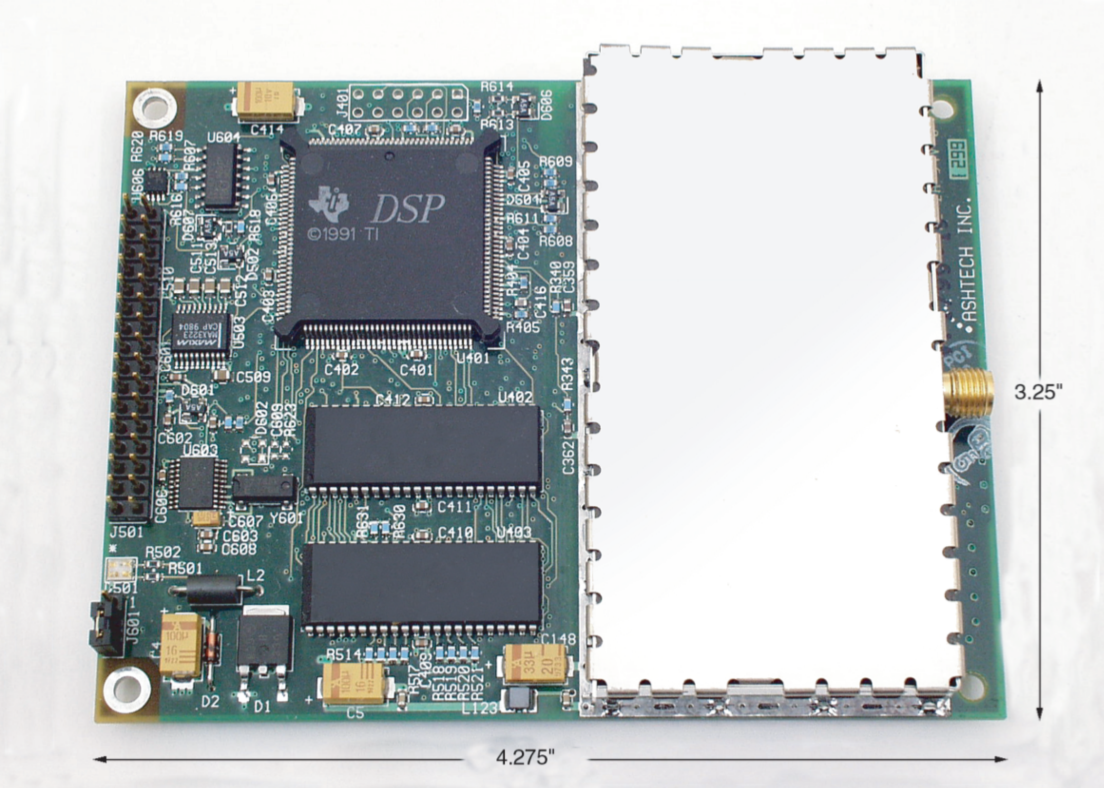


Figure 2.2. ***Thales SkyNav GG12 Receiver***: *In image of the Thales SkyNav GG12 Receiver from its technical manual [20].*

the positions of the SVs and determine the health of each SV. Ephemeris data is valid for two hours after it is generated. Almanac data is used to validate the position of the SVs that is calculated from the ephemeris.

2.1.1 GG12 Receiver

The Thales SkyNav GG12 is a GPS receiver that supplies the OU LAAS base station with the necessary raw data and messages used in the GBAS algorithm. The GG12 meets the DO-178B requirements and it monitors GPS and GLONASS constellations on the L1 frequency band [3]. Figure 2.2 shows a GG12 receiver.

2.1.1.1 Hardware Interface

The GG12 is equipped with a single J501 dual inline (DIN) header connector for data and power connections. The GG12 receives signal from an antenna via a 50Ω coaxial sub-miniature version A (SMA) connector. Each GG12 is powered via an external $5V_{DC}$ supply. The J501 connector has two bidirectional RS232 communication ports that are used to configure the GG12 for operation and to receive the GPS data. A preexisting custom printed circuit board (PCB) was used that allowed for connections to the J501 connector via different connections. Each RS232 communication port is connected to a female DB9 connector and each power input is connected to a Molex connector. The PCB includes a connector for a coin cell battery used for memory and real time clock backup, LEDs that are used as indicators for the GG12 status signal, and a fuse on the power circuit. The PCBs have also been modified such that the 1 pulse per second signal (1PPS) is also available for use with a VDB transmitter as described in section 2.4. Figure 2.3 shows a diagram of the PCB used with the GG12 receivers [21].

2.1.1.2 Receiver Communication

Each GG12 is configured in the same OU LAAS standard operating configuration by first resetting the receiver, secondly confirming the health and communication of the receiver, and thirdly by configuring the receiver for continuous output of measurement data on one of the two data ports. The communication is performed via RS232 communication protocol. Table 2.1 describes the communication protocol parameters used in OU LAAS. The default data rate for a GG12 receiver is 9600 baud but as part of the receiver configuration the baud

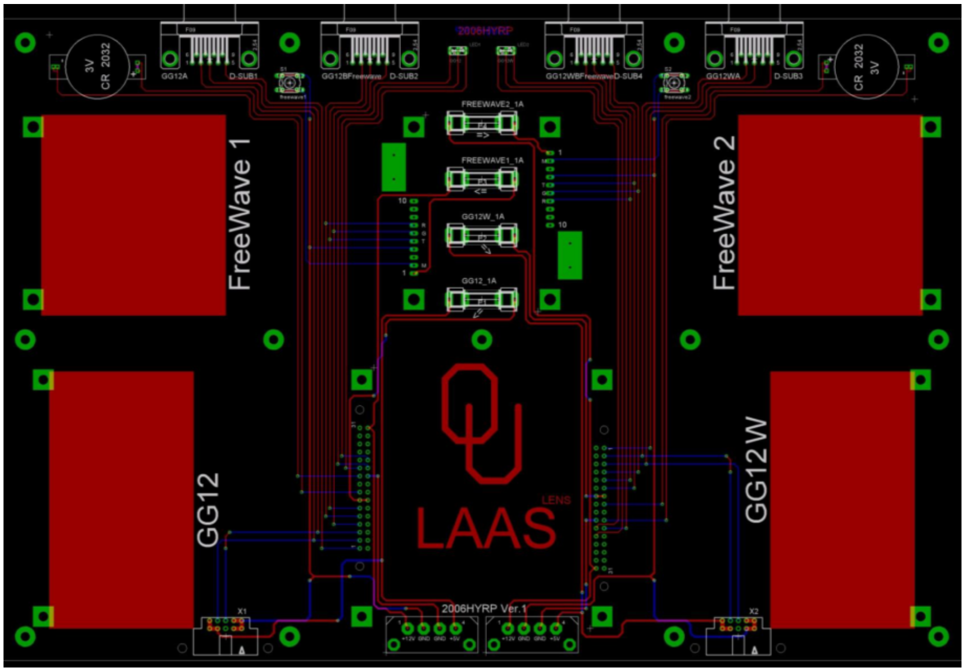


Figure 2.3. ***GG12 Printed Circuit Board**: Circuit diagram of preexisting custom printed circuit board used to power and interface with GG12 receivers [21].*

rates are set to 115200 baud, the fastest rate available.

Table 2.1. ***GG12 Communication Protocol Parameters**:*

Parameter	Value
Baud Rate	9600 or 115200
Data Bits	8
Stop Bits	1
Parity	None

The GG12 can output data in either binary or ASCII formats and receives commands in ASCII format. Regardless of the message format, every GG12 message begins with a header that contains "$PASH" followed by a character that defines the type of message used in the communication. Table 2.2 defines the different headers available in GG12 messages. After each header is an ASCII command code that defines specifically what data will be passed in the message body and what action will be taken by the receiver. The header, command code and message body are all comma delimited. The bodies of messages that use ASCII format have comma delimited fields and bodies of messages that use binary format do not use delimiters but use predefined bit structures.

Table 2.2 GG12 Message Headers	
Message Header	Message Type
$PASHQ	Send Query to GG12
$PASHS	Send Set Command to GG12
$PASHR	Receive Response from GG12

Table 2.2. ***GG12 Message Headers**: GG12 message headers and their meanings [20].*

Five GG12 raw data formats are used to support the data messages required for the GBAS algorithm. Table 2.3 lists the raw message outputs used. All of the messages used to receive raw data follow unique binary formats. The MCA, MIS and PBN messages are sent periodically at 2 Hz. The SAL and SNV messages are sent only when new almanac and ephemeris information are available. The GG12 communication algorithms are defined in section 3.2.

Table 2.3 GG12 Raw Messages		
Message Code	Message Header	Message Description
SAL	$PASHR,SAL,	Almanac
SNV	$PASHR,SNV,	Ephemeris
MCA	$PASHR,MCA,	Raw Measurements

Table 2.3 (*cont*)		
MIS	$PASHR,MIS,	Miscellaneous Raw Measurements
PBN	$PASHR,PBN,	Raw Position Data

Table 2.3. ***GG12 Raw Messages***: *GG12 raw data messages used.*

2.2 GPS Antennas

Novatel GPS-702-GG antennas are used to receive the GPS signals used by the GG12 receiver. The GPS-702-GG is a pinwheel antenna designed with integrated band-pass filters around GPS L1 and L2 frequencies [22]. The GPS-702-GG has an integrated amplifier with 27 dB of gain and noise figure of 2.5 dB. Figure 2.4 shows one of the GPS-702-GG antennas inside of an enclosure designed to mitigate multipath signal interference [23].

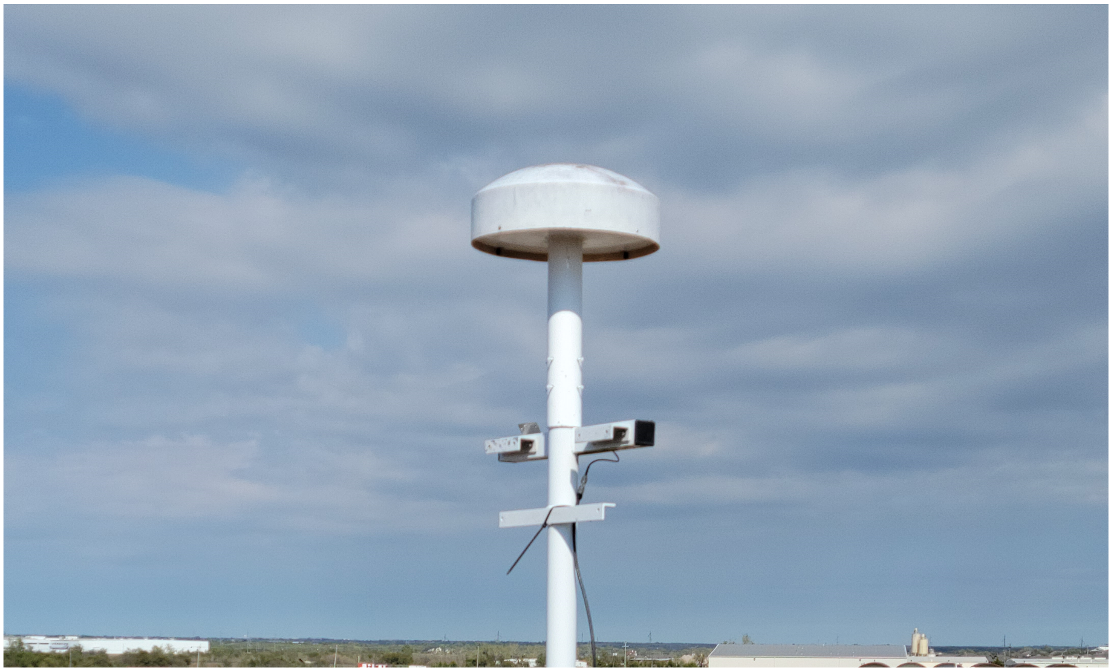


Figure 2.4. ***GPS Antenna***: *An image of the GPS-702-GG antenna installed on the roof of the LGF inside a multipath mitigating enclosure.*

The antennas are powered by the GPS Networking NALDCBS1X2 GPS Amplified Splitter. The splitter provides 22 dB of gain and a 12V supply to the antenna [24]. One output of the splitter is attached to the input of the GG12 receiver. Figure 2.5 shows the locations of the GPS antennas used.

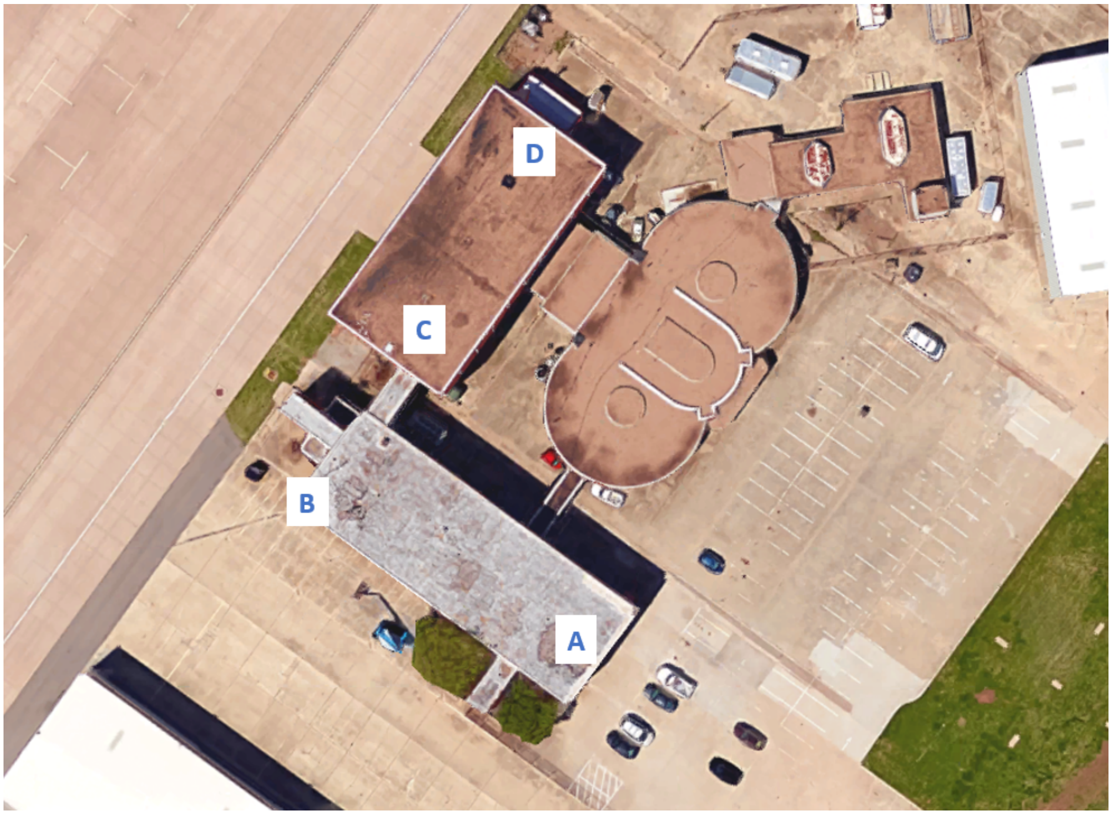


Figure 2.5. ***Antenna Locations:*** *An image of the roof of the OU GBAS facility with the locations of the GPS antennas marked A, B, C, and D.*

2.3 Computer Interface

The LVBS 3.0 computer has an Intel Xeon CPU 3.4 GHz processor with 32 GB of RAM and runs Windows 10. The software is written in LabVIEW 2020 SP1 (64-bit). Two 1x4 USB hubs are used to provide additional USB ports that are populated with USB to RS232 serial adapters that are used to interface with the GG12 receivers. Two additional USB to RS232 serial adapters are used to interface with the VDB transmitter and receiver. Figure 2.6 shows a hardware connection diagram that the LVBS 3.0 software was developed and runs with.

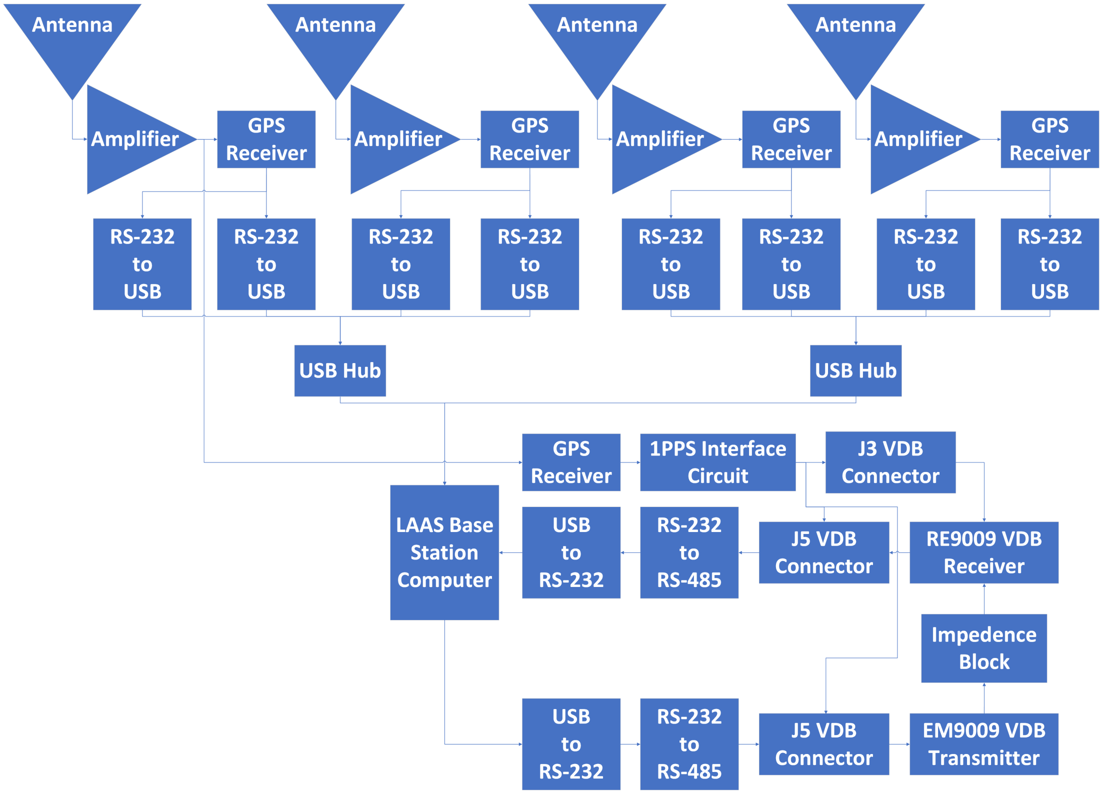


Figure 2.6. *OU LAAS Connection Diagram: Connection diagram of the OU LAAS workstation built to develop the LVBS 3.0 software.*

2.4 VHF Data Broadcast

Correction data are transmitted from the LVBS 3.0 system using the Telerad EM9009 VHF Data Broadcast (VDB) transmitter. The EM9009 transmitter was designed for use with LAAS and has an operating frequency of 108.025-117.95 MHz [25]. The EM9009 output is monitored from the Telerad RE9009 VDB receiver. Custom cabling was built to connect the EM9009 and RE9009 to the

LVBS 3.0 computer and to connect the one pulse per second (1PPS) signal to the systems. The 1PPS signal is acquired from a GG12 J501 connector and a custom converter was built to convert the signal to the RS485 signal required by the EM9009 and RE9009.

The RE9009 communicates using the J3 Remote Control connector and the J5 Data connector [26]. The EM9009 only requires a connection using a J5 Data connector to connect to the LVBS 3.0 computer. The J3 Remote Control connector is a DB-25 connector that is used to supply the 1PPS signal to the RE9009. Table E.1 in appendix E defines the pinout of the J3 Remote Control connector. The J5 Data connector is a DB-25 connector that the LVBS 3.0 computer uses to communicate with the EM9009 and RE9009 via RS485 communication. The 1PPS signal is also provided by the J5 Data connector. Table E.2 in appendix E defines the pinout of the J5 Data connector. An RS232 to RS485 adapter is used to connect from the LVBS 3.0 USB to RS232 adapter to the custom cabling used to connect to the J5 Data connector.

Chapter 3

Software Architecture Design Decisions and Implementation

The prototype LabVIEW implementation of CL-GBAS that was created in 2007 was carefully examined prior to development of the software demonstrated in this thesis. The prototype software had an architecture in which data acquisition and algorithm performance all occurred in series in a single processing thread. The combined data acquisition and algorithm thread would fall behind and miss samples. Only errors were logged and logs were formatted as comma separated values (CSV) files. The GG12 receiver was the only receiver that was used in the prototype so the communication and algorithm were interwoven in such a way that integrating a different receiver would have required a large amount of reworking. VDB transmission was partially implemented but was incomplete. This work develops the LabVIEW Base Station 3.0 (LVBS 3.0) software that improves on the prototype software by implementing a new software architecture, new data structures implemented with maps and sets, and new classes specifically developed in this work to allow the OU LAAS system to run in real time and function as a foundation for future experiments with GBAS and CL-GBAS.

3.1 Developing the Data Logger Class to Enhance Data Portability

The current work developed the Data Logger class to provide a flexible way to log data at every step along the data pipeline in a portable format. The JavaScript Object Notation (JSON) was chosen as a data log format because it is a common and human-readable format that is flexible enough to support any LabVIEW data types as name-value pairs. Instances of the Data Logger class are used in several locations within each process to log data related to the state of the process and the process errors that are generated. The Data Logger instances were implemented to be configurable so that they can be adjusted without software changes.

Classes used in object-oriented programming in LabVIEW are restricted to only contain private data. Data for each class are passed by-value, meaning a single copy of the data is available for each thread that accesses it (multiple copies may be made but they are all unique in memory). References can be added to private class data to allow multiple threads to access the same data in memory. The Data Logger class uses private data and internal private reference to a lossless queue to collect data from several threads. The data in the queue are logged in a single thread per log file, preventing conflicts when the software tries to access a file to write the data to local storage. A single Data Logger instance is created for each file that will be written. The architecture that was developed involving the reference to the lossless queue was chosen because it allows data to be written in a separate thread without blocking the execution of the thread from which the data are generated.

The Data Logger object is first initialized with a base file name, a directory, a maximum file size, and a maximum queue length. A queue reference is created when the object is initialized. After the Data Logger is initialized it may be copied into multiple threads where logger subVIs can be used to enqueue data onto the shared queue reference related to that given file. When data are enqueued they are converted from any native LabVIEW data type (or clusters composed of native LabVIEW data types) to JSON and inserted into the file as a JSON array of objects. Malleable VIs (VIMs) were developed to allow data of any type to be input and converted to JSON. Data are flushed from the queue and written to local storage periodically. Data will also be flushed to disk if the number of elements in the queue exceeds the maximum limit set during creation. Files are created using the base file name defined during initialization appended with a timestamp of when the file was first created. A new file with the base file name and new timestamp is created when the current file exceeds the maximum file size specified during initialization.

3.2 Developing the Receiver Class as a Hardware Abstraction Layer

The Receiver class developed in this thesis was designed to make integrating different types of GPS receivers in the OU LAAS system straightforward.

Though the prototype base station software was designed around a single receiver type, advances in technology and future research requirements demand that a more flexible system be developed to integrate different GPS receivers. The Receiver class library was designed and implemented to function as a hardware abstraction layer (HAL), a system that makes interfacing with different pieces of hardware the same from a high level software perspective. The Receiver class library contains the parent class Receiver and child classes that function as specific implementations of different GPS receiver. The LVBS 3.0 software creates instances of the subclasses of the Receiver class that inherit the same high level subVIs from the Receiver class. The Receiver class subVIs are used to call the subVIs that are specific to each child subclass as needed during runtime. This software design decision allows for more child subclasses to be added to the Receiver class library without modifying the structure of the Data Acquisition process. The GG12 Receiver class is a fully implemented child class of Receiver and is used to interface with a Thales SkyNav GG12 in LVBS 3.0. Several other child classes have been created and partially implemented. Figure 3.1 shows the LabVIEW class hierarchy for the Receiver class and the child classes.

The Hot Start script is an optional segment of code that can be configured to run at the start of operation of LVBS 3.0. The script retrieves the current state of almanac and ephemeris data from a receiver so that the algorithm can begin with it instead of waiting for data to come in over time, reducing the amount of time required before LVBS 3.0 is ready to transmit. Hot Start can be configured to use different GPS receivers using the Receiver class. Hot Start is run before the main processes begin so that the DAQ process and Hot Start can use the

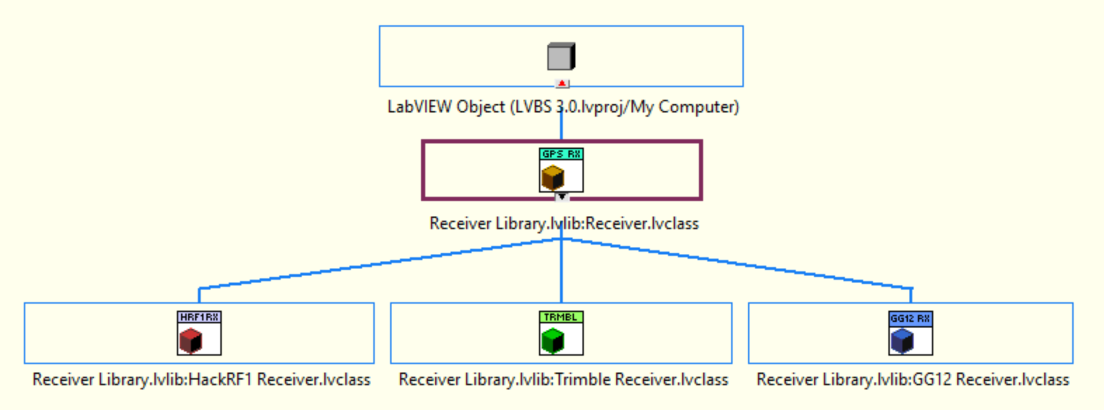

LabVIEW Object (LVBS 3.0.lvproj/My Computer)
GPS RX
Receiver Library.lvlib:Receiver.lvclass
HRF1RX
Receiver Library.lvlib:HackRF1 Receiver.lvclass
TRMBL
Receiver Library.lvlib:Trimble Receiver.lvclass
GG12 RX
Receiver Library.lvlib:GG12 Receiver.lvclass

Figure 3.1. ***LabVIEW Receiver Class Hierarchy***: *The Receiver class inherits from the class Object. The child classes GG12 Receiver, Trimble Receiver and HackRF1 Receiver inherit from the Receiver class.*

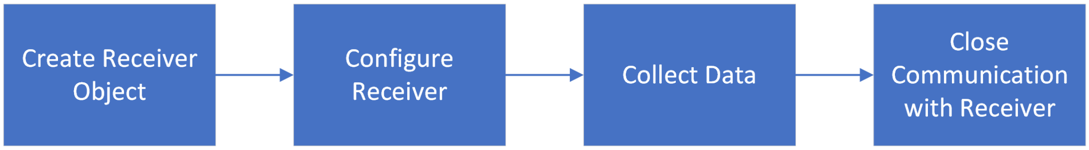


Figure 3.2. ***Hotstart Script Operations***: *When Hot Start is enabled, the script creates a Receiver of a child class as defined in the configuration, configures the GPS receiver according the the implementation in the given child class, collects the data, and stops communication with the receiver so that it may be used for other processes later in the program.*

same receiver without conflicting. Figure 3.2 shows the order of events run during the script if Hot Start is enabled.

3.2.1 Receiver Subclass Implementation for GG12 GPS Receiver

The GG12 GPS receiver was the first hardware implemented as a subclass to the Receiver class. Each subclass must implement the subVIs used to collect GPS data including almanac, ephemeris, health and pseudorange measurements that are used in normal operation of LVBS 3.0. Receivers may optionally implement the subVIs used to run the Hot Start script that runs before normal

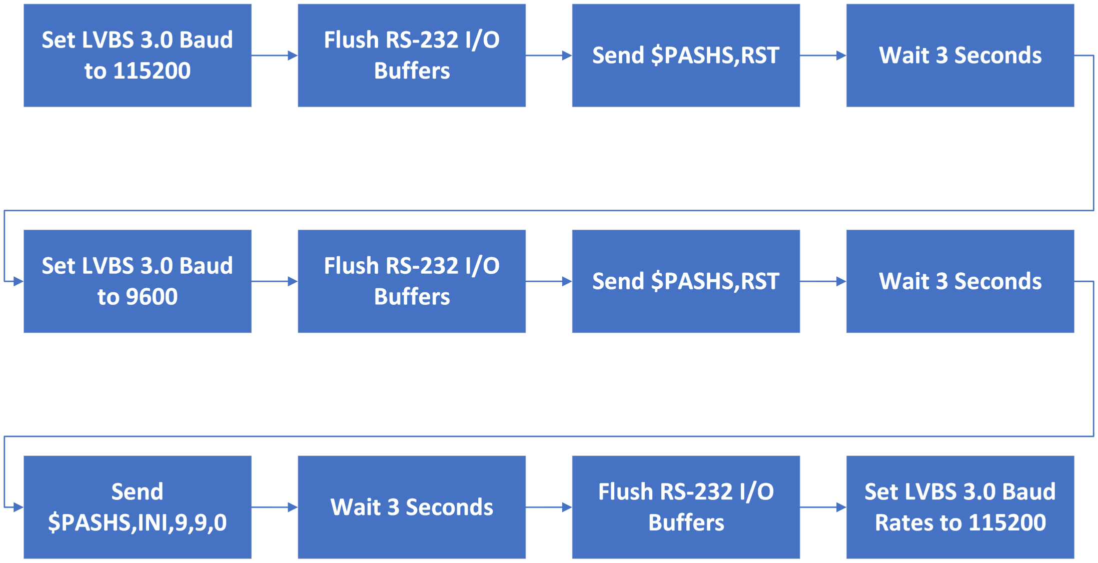


Figure 3.3. *GG12 Receiver Configuration Script Diagram: $PASHS,RST is sent at both 115200 baud and 9600 baud because 115200 is the standard operating baud rate for the GG12 in LVBS 3.0 but 9600 is the default setting for a GG12. The RS-232 serial communication buffers are reset between reset commands.*

operation of LVBS 3.0.

The GG12 Receiver subclass is initialized by resetting the GPS receiver to clear the previous configuration. The $PASHS,RST command is sent at 115200 baud and 9600 baud to the debug RS232 port to reset the GG12 to the default settings. The $PASHS,INI,9,9,0 command is used to set the data rate on both of the RS-232 ports of the GG12 to 115200 baud without clearing the stored ephemeris and almanac data from the GG12. There is a three second delay between each command to give the GG12 enough time to perform the functions. Figure 3.3 depicts the order of operations used to reset the GG12.

After the GG12 has been reset, the receiver is configured for standard operation by using the commands listed in table 3.1 or configured for use with Hot Start using the commands listed in table 3.2. During standard operation, the

GG12 receiver is configured for periodic transmission of data. When configured for Hot Start the GG12 receiver is not configured for periodic transmission. This allows for the required data to be available but only when queried.

Table 3.1 GG12 Configuration for Normal Operation

	GG12 Command	Function
1	$PASHS,RST	Resets the GG12 to factory settings
2	$PASHS,POP,2	Set internal receiver update rate to 2 Hz
3	$PASHS,RCI,2	Set serial data output rate to 2 Hz
4	$PASHS,SYS,GPS	Set reveiver to GPS constellation only
5	$PASHS,ION,N	Disable ionospheric model error correction
6	$PASHS,TRO,N	Disable tropospheric model error correction
7	$PASHS,ELM,1	Set elevation mask to only allow satellites above 1°
8	$PASHS,MSV,1	Set minimum number of SVs for serial data output to 1
9	$PASHS,PEM,5	Set elevation mask for position solution to 5°
10	$PASHS,USP,GLO,N	Disable use of all GLONASS SVs
11	$PASHQ,RID	Query receiver settings
12	$PASHQ,RWO,A	Query raw data output settings on port A

Table 3.1 (*cont*)		
13	$PASHS,RAW,MCA,A,ON,.5	Enable 2 Hz MCA output message on port A
14	$PASHS,RAW,MIS,A,ON,.5	Enable 2 Hz MIS output message on port A
15	$PASHS,RAW,PBN,A,ON,.5	Enable 2 Hz PBN output message on port A
16	$PASHS,RAW,SAL,A,ON,.5	Enable 2 Hz SAL output message on port A
17	$PASHS,RAW,SNV,A,ON,.5	Enable 2 Hz SNV output message on port A
18	$PASHQ,RWO,A	Query raw data output settings on port A
19	$PASHS,SPD,A,9	Set baud rate on port A to 115200 baud
20	$PASHS,SAV,Y	Save new settings

Table 3.1. ***GG12 Configuration for Normal Operation****: Commands used to configure a GG12 for normal operation.*

Table 3.2 GG12 Configuration for Hot Start		
	GG12 Command	Function
1	$PASHS,RST	Resets the GG12 to factory settings
2	$PASHS,POP,2	Set internal receiver update rate to 2 Hz
3	$PASHS,RCI,2	Set serial data output rate to 2 Hz
4	$PASHS,SYS,GPS	Set receiver to GPS constellation only
5	$PASHS,ION,N	Disable ionospheric model error correction

Table 3.2 (*cont*)

6	$PASHS,TRO,N	Disable tropospheric model error correction
7	$PASHS,ELM,1	Set elevation mask to only allow satellites above 1°
8	$PASHS,MSV,1	Set minimum number of SVs for serial data output to 1
9	$PASHS,PEM,5	Set elevation mask for position solution to 5°
10	$PASHS,USP,GLO,N	Disable use of all GLONASS SVs
11	$PASHQ,RID	Query receiver settings
12	$PASHQ,RWO,A	Query raw data output settings on port A
13	$PASHS,RAW,MCA,A,OFF,.5	Disable 2 Hz MCA output message on port A
14	$PASHS,RAW,MIS,A,OFF,.5	Disable 2 Hz MIS output message on port A
15	$PASHS,RAW,PBN,A,OFF,.5	Disable 2 Hz PBN output message on port A
16	$PASHS,RAW,SAL,A,OFF,.5	Disable 2 Hz SAL output message on port A
17	$PASHS,RAW,SNV,A,OFF,.5	Disable 2 Hz SNV output message on port A
18	$PASHQ,RWO,A	Query raw data output settings on port A

Table 3.2 (*cont*)		
19	$PASHS,SPD,A,9	Set baud rate on port A to 115200 baud
20	$PASHS,SAV,Y	Save new settings

Table 3.2. ***GG12 Configuration for Hot Start****: Commands used to configure a GG12 for Hot Start.*

During standard operation, data collection using the GG12 Receiver subclass is performed by listening to an RS-232 serial channel until the periodic data stream begins. Once the data start coming in, they are captured and concatenated in memory. Data are expected every 500 milliseconds so a timeout error was implemented if data are not collected for more than five seconds. Once the data have stopped being collected for more than 50 milliseconds, the data collection stops and the data are parsed and normalized into the LVBS 3.0 data structures. Figure 3.4 shows the data collection algorithm for the GG12 receiver.

During the Hot Start script, data are collected from the GG12 receiver by querying the receiver for the almanac and ephemeris. LVBS 3.0 records data after each query and waits until no data have been received for 1 second to verify that each query is complete. Responses from each query are broken into several transmissions that are sent every 500 milliseconds. After both queries are complete data are processed and converted from GG12 binary format to LVBS 3.0 standard data structures to be used in the algorithm. Figure 3.5 depicts the algorithm used for querying the GG12.

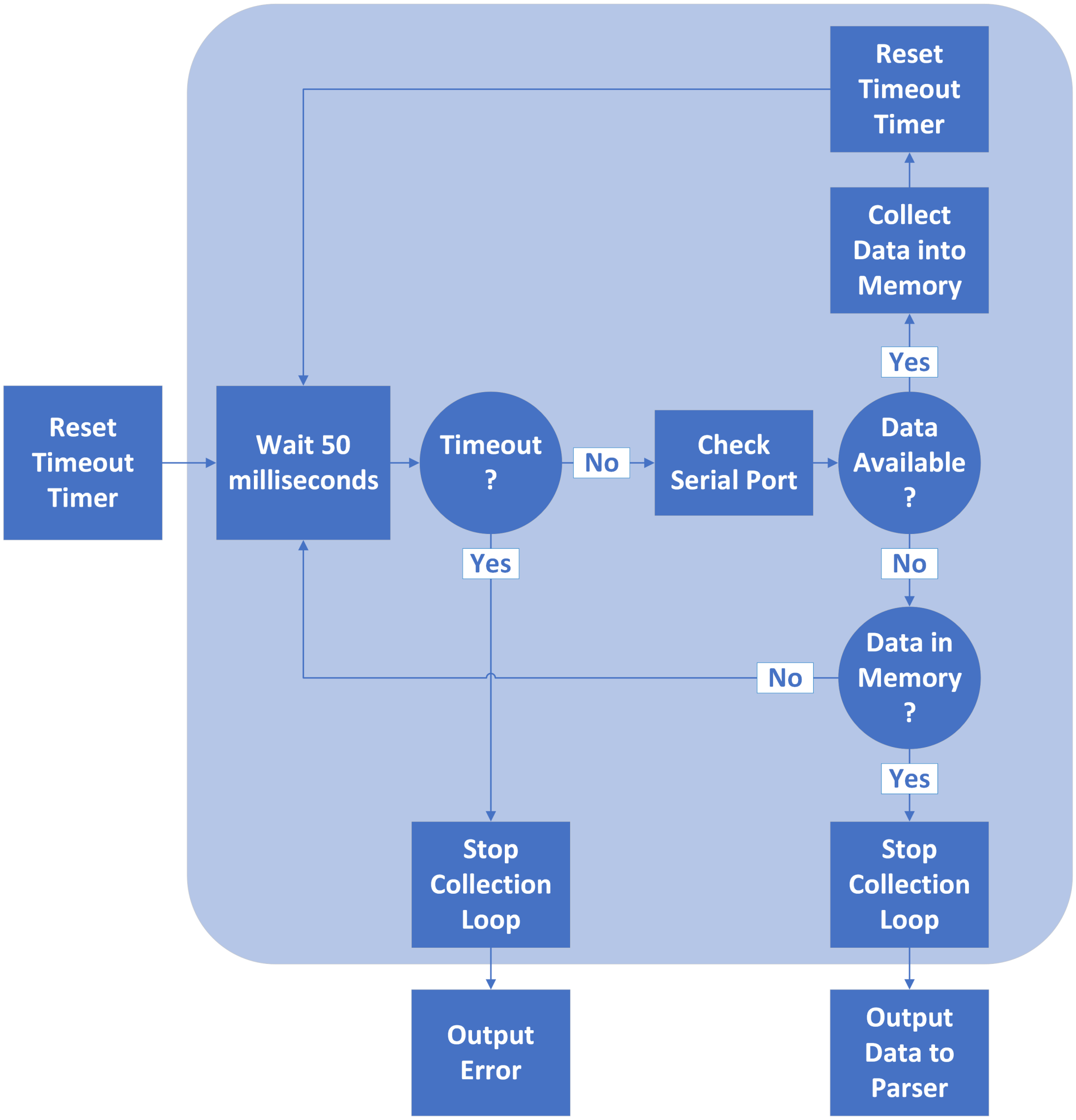


Figure 3.4. ***Data Collection Algorithm Diagram***: *Data collection is performed by waiting for a data transmission to begin, collecting the data into memory, and then sending data to be parsed once data collection has ended. Collection is considered failed if no data are collected within 5 seconds of the beginning of an attempt. The GG12 is configured to output data messages every 500 milliseconds.*

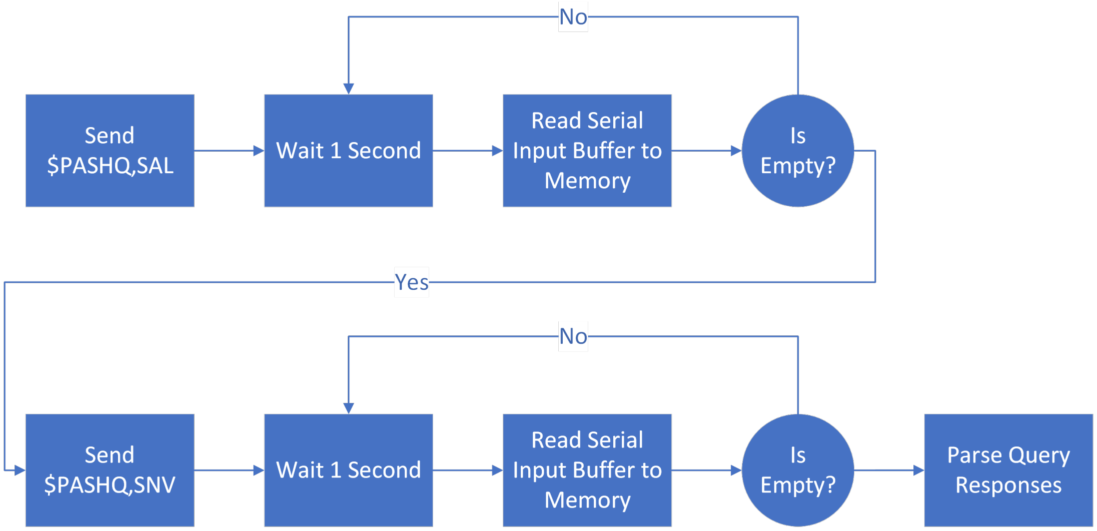


Figure 3.5. ***Almanac and Ephemeris Queries:*** *Queries for almanac and ephemeris data are performed in series before being processed into native LabVIEW data types.*

3.3 Developing the Data Flow Architecture

The LVBS 3.0 software was developed using parallelism and pipelining techniques to fulfill the GBAS timing requirements. The inability of the prototype software to meet the timing requirements of GBAS was a central focus of the development presented in this thesis. Pipelining is a software technique that uses multiple threads to perform a series of actions in order and in parallel on a set of data allowing for greater efficiency. Multithreading is a software technique that allows several processes to occur in parallel without blocking each other. The Data Acquisition (DAQ) process was implemented using parallelism to allow communication with each GPS receiver to occur simultaneously on separate threads. The DAQ and Algorithm processes were implemented separately according to a producer-consumer relationship where the DAQ process can send the latest data to the Algorithm process without blocking the DAQ from receiving the latest set of data. The producer-consumer implementation of pipelining allows for greater time efficiency of the system. The Type Message and VDB

Transmit processes were created to format the GPS correction data into Type 1 messages and send them to an EM9009 VDB for transmission.

The LVBS 3.0 software follows a startup script that reads configuration data for each of the main processes, initializes the graphical user interface (GUI), initializes each of the main processes, and runs the Hot Start script if configured to do so. This ensures that each of the main processes begin together and the Hot Start script allows the algorithm process to generate corrections earlier. The main processes developed in this thesis are the DAQ process, Algorithm process, Type Message process, and VDB Transmit process. They are organized into a series of producer-consumer data flow relationships that allows them to meet each of the different timing requirements of GBAS. The Settings process saves changes to the settings automatically when they are configured during operation. The Update Status process updates the GUI with data from the other processes. Figure 3.6 is a diagram showing the software architecture at a high level.

The LVBS 3.0 software uses producer-consumer relationships between the DAQ, Algorithm, Type Message and VDB Transmit main processes that pass data via tag or stream channels. Tag channels function as lossy single element queues that only store the latest single copy of data in memory. Stream channels function as multiple element queues that can be used to maintain a continuous history of data to be processed so that no elements are lost. When data are read from a tag channel, the queue remains filled with the existing value. When data are read from a stream channel the element that was read is removed from the queue. When a channel stream is empty the consumer of the data will

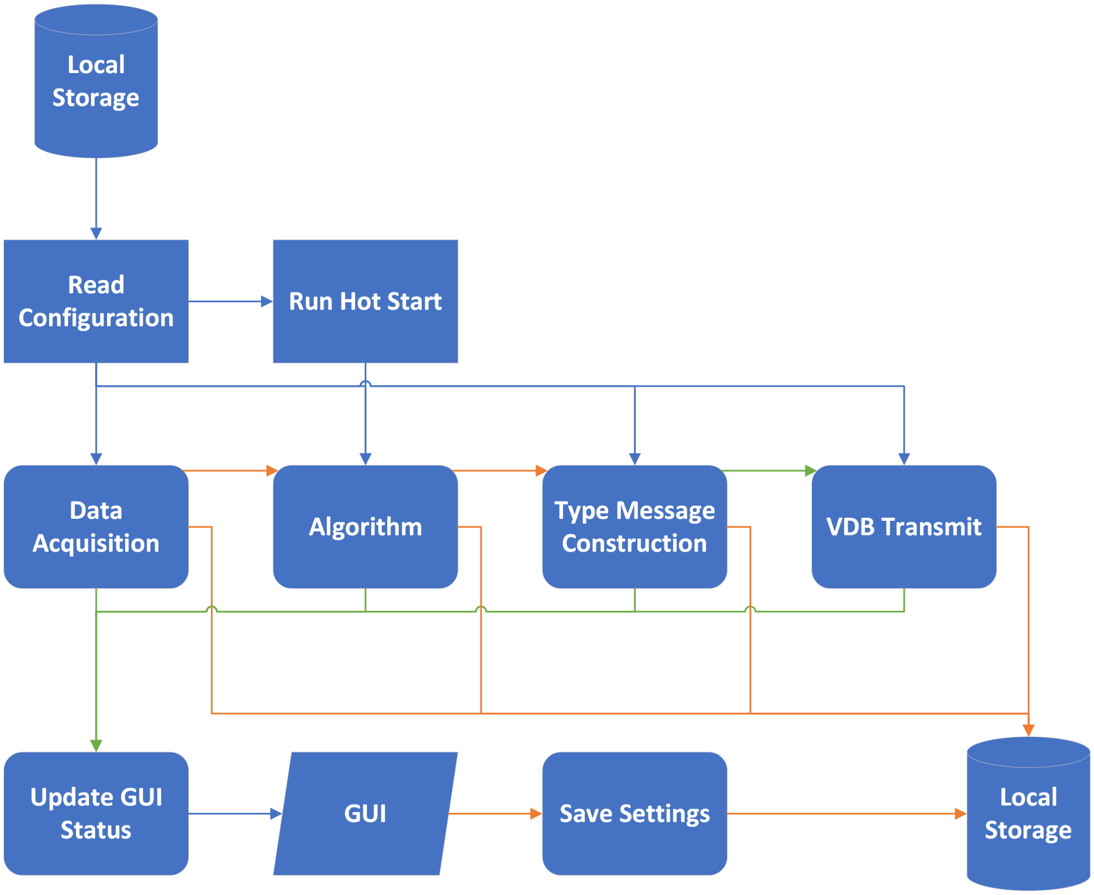


Figure 3.6. ***Software Architecture****: Configuration and settings data are read from local storage then, if necessary, the Hot Start script is performed. The six main processes are started in parallel, each logging to local storage and updating the status in the GUI. Blue wires indicate direct data flow by value, green wires indicate lossy single element channels and orange wires indicate lossless channel streams and queues.*

wait until a timeout before continuing without data, blocking execution of the consumer thread until data are available. Within each of the main processes data are passed to several multiple element queues that hold the error and state data and are periodically converted to JSON and logged to the local storage. Memory leaks occur when producers fill queues and channels faster than consumers can process the data. Memory leaks are avoided using size limits to the queues and channels.

3.3.1 Data Acquisition Process

The DAQ process is made up of five cloned instances of the DAQ Receive subprocess and the DAQ Sync subprocess each running in parallel. The DAQ Receive subprocesses collect, parse, and log data from the GPS receivers and output the collected data to the DAQ Sync subprocess. The DAQ Sync subprocess collects the data from all of the receivers and synchronizes them in time before sending the latest available data to the Algorithm process. Figure 3.7 shows the flow of data between subprocesses inside the DAQ process.

3.3.1.1 DAQ Receive Subprocess

The DAQ-Receive subprocess is made up of one thread that handles communication with the GPS receiver and parsing of receiver data, one optional thread that logs the output data of each receiver, and one optional thread that logs errors from the subprocess. Each copy of the DAQ Receive subprocess that is running occupies its own space in memory using preallocated clone reentrant execution. The communication with the GPS receiver and parsing of receiver

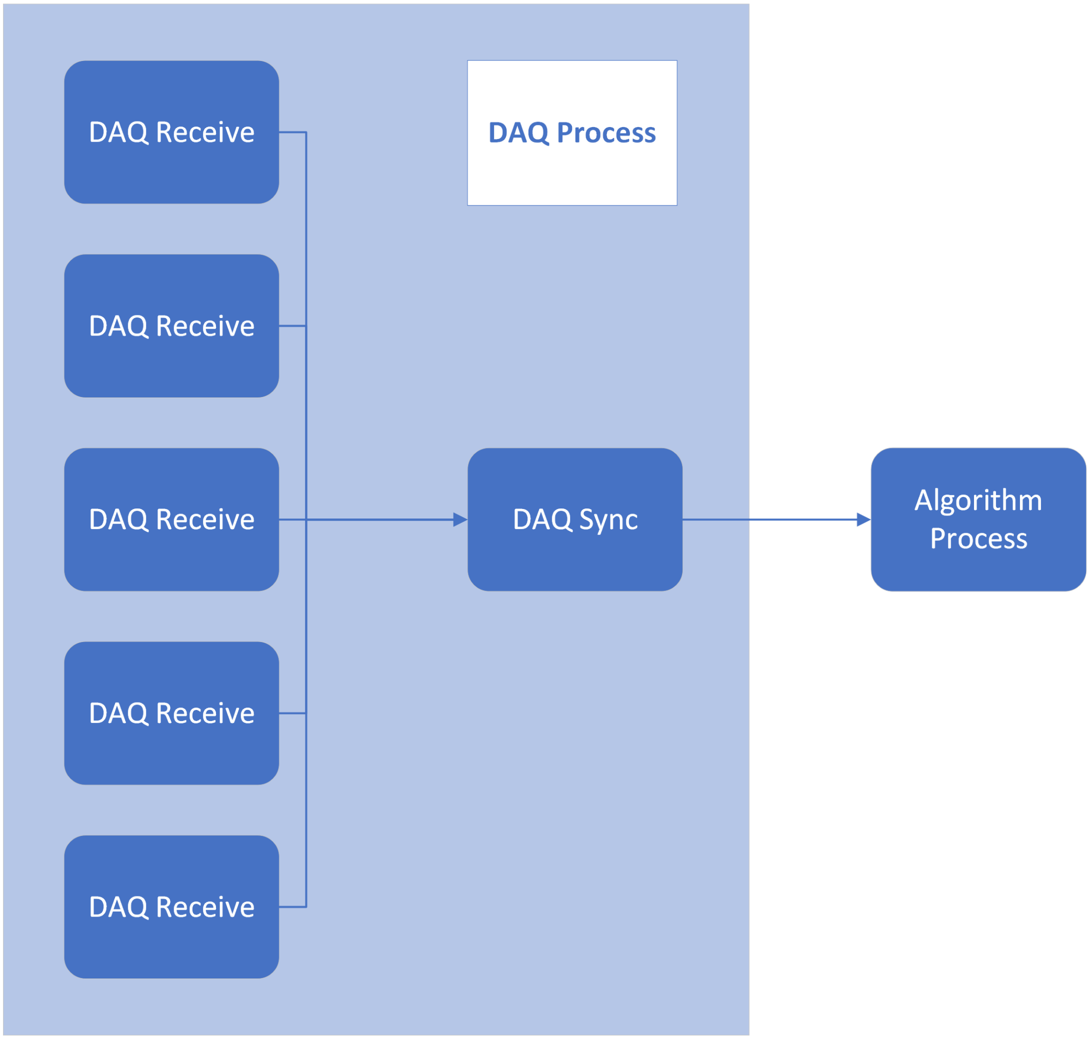


Figure 3.7. ***DAQ Process Diagram**: One DAQ Receive subprocess is created for each GPS receiver being used. All data from the receivers are parsed and sent to the DAQ Sync subprocess to get synchronized. A single set of data containing the data from each GPS receiver is output to the Algorithm process.*

data is performed using the Receiver class and subclasses. The output data and error log threads are performed using the Data Logger class.

The DAQ receive communication was designed to maintain good connection to the GPS receiver and to ensure that if the GPS receiver is disconnected it will be reconfigured for desired function. The GPS receiver communication begins by creating a Receiver class object that will be used to perform the communication. The Receiver object then initializes the GPS receiver for periodic communication. After the Receiver instance is initialized, the thread enters a loop that repeatedly waits to collect the data from the GPS receiver. Data are sent periodically to be logged and to the DAQ Sync subprocess via a lossy single element tag channel. If an error occurs during data collection or during data parsing the connection to the GPS receiver is closed, the Receiver object is destroyed and receiver communication starts over from creation of a Receiver object. Figure 3.8 depicts the receiver communication algorithm used to maintain a good connection to a GPS receiver.

3.3.1.2 Data Sync Subprocess

The purpose of the Data Sync subprocess that was developed in this thesis is to ensure that data from the GPS receivers share the same GPS receive times. This subprocess is critical because it ensures that the data from the RSs meet the GBAS timing and synchronization requirements while not necessarily requiring GPS receivers to return data at the same rates. This allows different GPS receiver types to be used together regardless of their communication cycle time. The Data Sync subprocess is made up of a single thread that collects data from

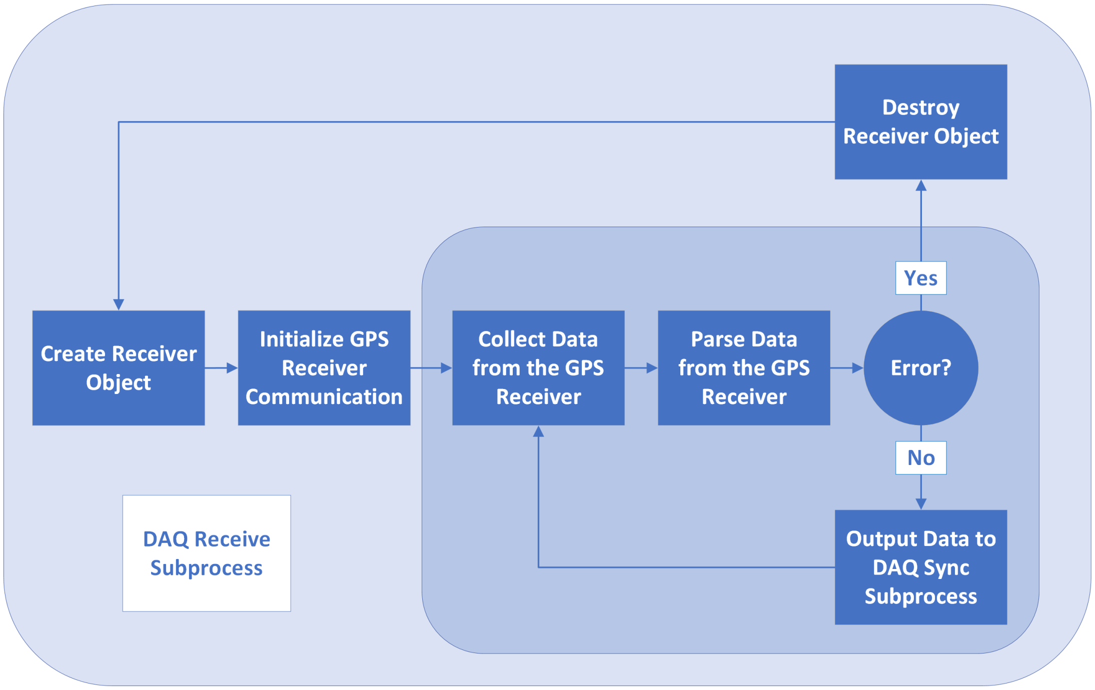

Destroy Receiver Object
Yes
Create Receiver Object
Initialize GPS Receiver Communication
Collect Data from the GPS Receiver
Parse Data from the GPS Receiver
Error?
No
DAQ Receive Subprocess
Output Data to DAQ Sync Subprocess

Figure 3.8. ***DAQ Receive Subprocess Diagram****: The DAQ Receive subprocess will attempt to reconnect and reset the GPS receiver until a connection is successfully established. Upon error it will try to stop it's current session and try to start a new one. The data collection step is described in figure 3.4.*

all of the Data Receive subprocess and performs the synchronization algorithm, one thread that logs the state of the subprocess, and one thread that logs errors generated by the subprocess. The state and error logging is performed using the Data Logger class.

Data are generated from each of the Data Receive subprocesses every 500 milliseconds and are enqueued onto a different lossy single-element tag channel for each Data Receive subprocess. The data synchronization algorithm begins by polling each tag channel for GPS data. The most recent five seconds of GPS data are maintained in memory for each of the tag channels. The GPS data are stored in a map structure that maps the receive time to the entire GPS payload bundled with the time the system last collected data with that given receive time. Sets of all receive times for each map are also maintained. Data are removed from the map when data for a given receive time have not been received for more than five seconds. Figure 3.9 shows the algorithm used to maintain a current set of data from the GPS receivers.

The intersection of all of the sets of receive times is calculated to determine that the set of data from all of the GPS receivers are from the same time. No data are sent to the Algorithm process if there is not a shared receive time from all of the GPS receivers. The most recent receive time in the set of shared receive times is stored in memory as the current receive time. GPS receive time is the amount of time, in milliseconds, since the beginning of the week. An algorithm was developed to determine the most recent receive time in the event of a weekly rollover. The algorithm determines if there are receive times that are before and after the middle of the week present in the set of available receive

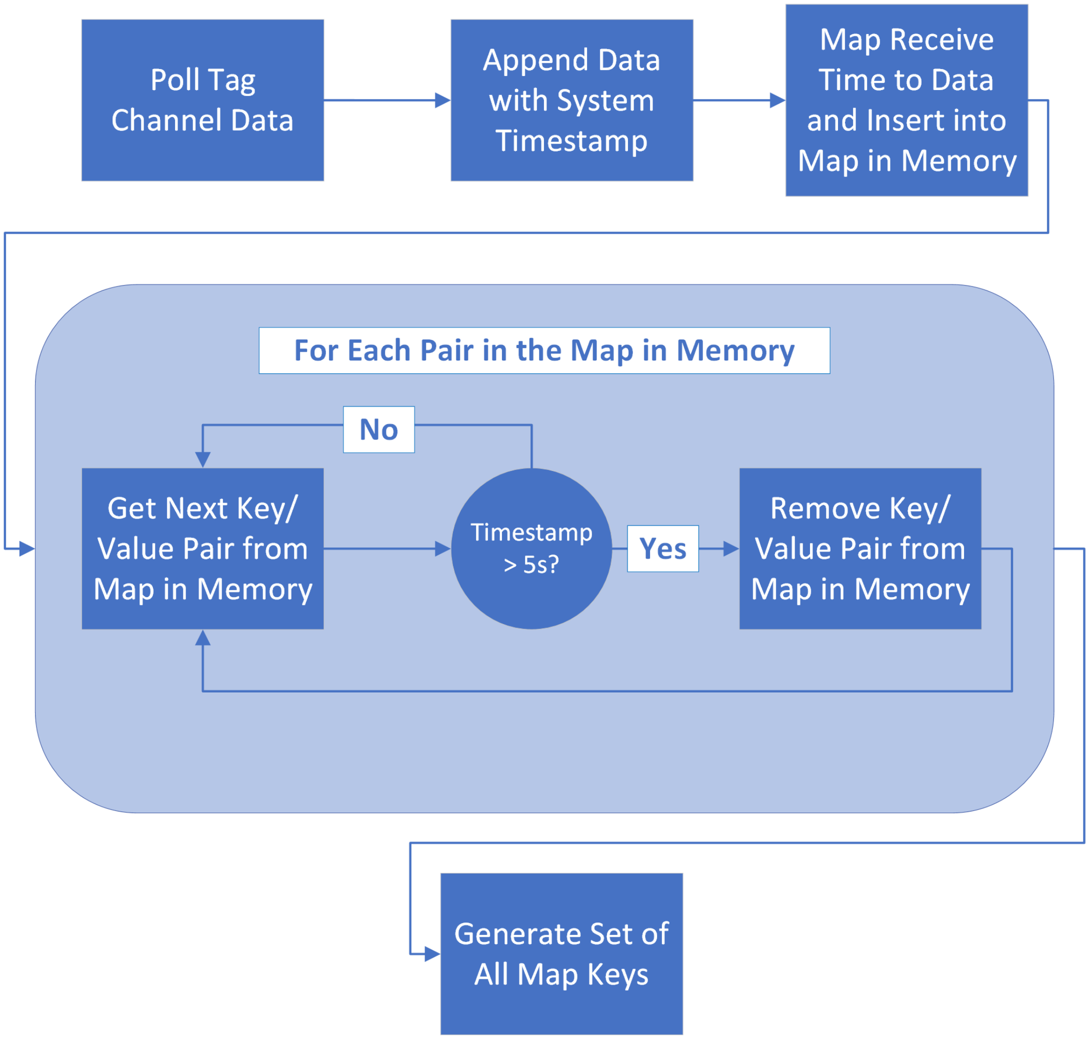

Poll Tag Channel Data
Append Data with System Timestamp
Map Receive Time to Data and Insert into Map in Memory
For Each Pair in the Map in Memory
No
Get Next Key/ Value Pair from Map in Memory
Timestamp > 5s?
Yes
Remove Key/ Value Pair from Map in Memory
Generate Set of All Map Keys

Figure 3.9. ***Data Synchronization Algorithm Diagram:*** *Data are mapped with the GPS receive time as the key to a cluster of GPS data from the receiver and the timestamp that the Data Sync subprocess received data with that GPS receive time. A set of receive times is generated from the keys in the map structure after the map is maintained for the current iteration of the Data Sync subprocess.*

times and, if there are, it excludes receive times after halfway through the week from the evaluation of the most recent receive time. The current receive time is used as a key to the five maps of five seconds of GPS data to retrieve most recent data with a shared receive time. Data are sent to the Algorithm process via a lossless stream channel. Data are only enqueued to the channel when the current receive time changes.

3.3.2 Algorithm Process

The Algorithm process created for this research performs calculations necessary to generate data that are used in the Type Message process. Data enter the Algorithm process via a lossless stream channel from the Data Acquisition process. The Algorithm process can optionally be initialized with the almanac and ephemeris data collected by the Hot Start script. The Algorithm process validates and maintains a current set of ephemeris data, calculates the positions of the SVs tracked in the ephemeris, calculates correction data, and calculates the error present after corrections have been applied to the LVBS 3.0 local monitor receiver. Data are passed to the Type Message process via a lossless stream channel. The algorithms used in LVBS 3.0 are built around map and set data structures and are implemented differently than those in the prototype software. Maps and sets were chosen as a foundation because many of the algorithms are designed to determine shared SVs between RSs or exclude certain SVs or RSs and such actions are straightforward and efficient when using maps and sets as opposed to arrays of structures in LabVIEW.

The raw SV validation information is stored in a pair of maps that map the PRN of the SV to the ephemeris and almanac information for each SV. These maps are updated when new almanac or ephemeris data is returned from the receiver that functions as the local monitor for LVBS 3.0. Every 500 milliseconds the satellite position algorithm updates the usable almanac and ephemeris data, calculates the source availability duration (SAD) of each SV, calculates the ephemeris decorrelation parameter of each SV, and calculates the position and elevation of each SV. The SAD that is calculated is required to be accurate within 60 seconds of the actual SAD. Appendix C defines in detail the calculations used to generate the SV information from the almanac and ephemeris data.

The Type 1 message is composed of three sections. Section 1 contains message data related to the time of validity of the Type 1 message, the additional message flag, the number of measurement blocks, and the type of measurement in the Type 1 message. Section 2 contains low frequency information related to the ephemeris used as a basis for the measurement block including the ephemeris decorrelation parameter, ephemeris CRC and the source availability duration (SAD) of an SV. A single Type 1 message only contains low frequency information about the single SV that is first in the measurement block. The low frequency information is sent for every SV that is used at least once every ten seconds unless the ephemeris information has changed. When new ephemeris information is used for the first time for an SV, the low frequency Type 1 message data for that SV is sent three frames in a row. Figure 3.10 depicts the transmission pattern of the low frequency data when new SVs are added to the ephemeris [1]. Section 3 contains high frequency data in the form of ranging

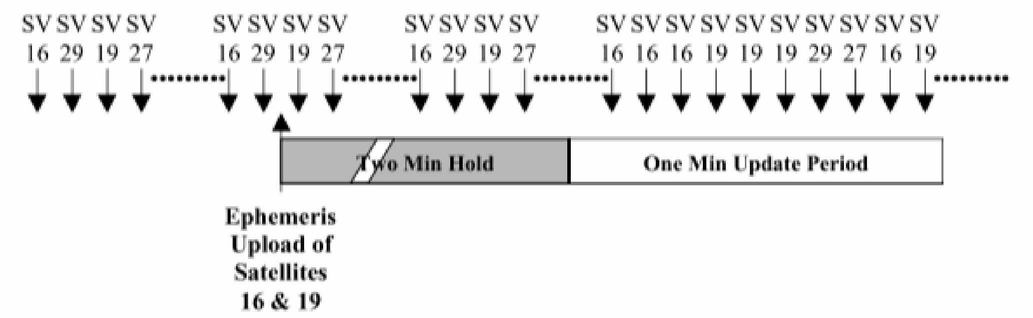


Figure 3.10. ***Ephemeris Broadcast Pattern****: Initially PRN 16, 29, 19, and 27 are available for broadcast and are transmitted in a rotation pattern accross multiple TDMA frames. The ephemeris information for PRN 16 and 19 is updated. After the two minute validation period the, the new ephemeris information is used as a basis for the measurement block. The first frames after the new ephemeris data is used includes the updated ephemeris for PRN 16 and 19 each repeating three times [1].*

source measurement blocks. Table 3.3 defines the data contents of a Type 1 message. Appendix D contains the calculations used to generate the data contents of the Type 1 message.

Table 3.3: Type 1 Message Structure

Description	Length (Bits)	Range	Resolution
Section 1: Message Information			
Modified Z-count	14	$0 - 1199.9s$	$0.1s$
Additional Message Flag	2	$0 - 3$	1
Number of Measurements	5	$0 - 18$	1
Measurement Type	3	$0 - 7$	1
Section 2: Low Frequency Data for One PRN			
Ephemeris Decorrelation Parameter	8	$0 - 1.275 \cdot 10^{-3}$	$5 \cdot 10^{-6} \frac{m}{m}$
Ephemeris CRC	16	-	-

Table 3.3: (*cont*)			
Source Availability Duration	8	$0 - 2540s$	$10s$
Section 3: High Frequency Data for N Measurement Blocks			
Ranging Source ID (PRN)	8	1 - 255	1
Issue of Data (IOD)	8	0 - 255	1
Pseudorange Correction (PRC)	16	$\pm 327.67m$	$0.01\ m$
Range Rate Correction (RRC)	16	$\pm 32.767\frac{m}{s}$	$0.001\frac{m}{s}$
$\sigma_{\text{PR\_GRND}}$	8	$0 - 5.08m$	$0.02m$
B1	8	$\pm 6.35m$	$0.05m$
B2	8	$\pm 6.35m$	$0.05m$
B3	8	$\pm 6.35m$	$0.05m$
B4	8	$\pm 6.35m$	$0.05m$

Table 3.3. ***Type 1 Message Structure***: *Each value is calculated for each SV that is available for broadcast.*

The CL-GBAS algorithms that are implemented are centered around calculating the error present before and after the Type 1 message corrections are applied to measurements taken from the LM receiver. The Vertical Error (VE) and Horizontal Error (HE) represent the distance in meters that a position solution differs from the surveyed position of the receiver antenna. The CL-GBAS LGF calculates the VE and HE based on the surveyed location of the LM receiver antenna and the corrected position vector of the LM [3]. The Vertical Protection Level (VPL) and Horizontal Protection Level (HPL) are terms that describe the risk of excessive error in the Type 1 corrections. CL-GBAS uses the VPL and HPL calculated from the corrected position of the LM to evaluate whether the Type 1 messages should be sent. Appendix D describes in detail how the CL-GBAS calculations are performed.

3.3.3 Type Message Process

The Type Message process prepares messages to be sent during the VDB Transmit process. This thesis develops the LAAS Type 1 message into the LVBS 3.0 software. The data that are used to populate LAAS message Type 1 come into the Type Message process via a lossless stream channel. LAAS messages are added to a Time Division Multiple Access (TDMA) formatted message as application data. The TDMA application data are formatted into a message that can be received by the Telerad EM9009 VDB.

3.3.3.1 TDMA Message Development

TDMA messages are divided into many 3-bit symbols. TDMA messages contain multiple forward error corrections (FECs), an error correction method that appends correction data to an outgoing message that can be used to detect and correct errors in the received data [27]. TDMA messages begin with a training sequence consisting of Power Stabilization, Synchronization and Ambiguity Resolution, Station Slot Identifier (SSID), Transmission Length, and a Training Sequence FEC. The training sequence is followed by application data consisting of a LAAS message, an Application FEC and additional fill bits if necessary. Fill bits are arbitrary bits that are added to ensure that all of the data in the transmission fits into 3-bit symbols. Bit scrambling is performed on the SSID, Transmission Length, Training Sequence FEC, Application Data, and Application FEC. Message symbols will be converted into differentially-encoded 8 phase shift keyed (D8PSK) carrier phase shifts during transmission.

The Power Stabilization segment is a 15 bit-sequence of all 0 values that is meant to give the transmitter time to signal power up and stabilize before the message is sent [1]. The Synchronization & Ambiguity Resolution segment is a 48-bit segment that "was designed to have good auto-correlation properties and to facilitate estimation of the center frequency" [1]. The sequence is always:

010 001 111 101 111 110 001 100 011 101 100 000 011 110 010 000

The SSID is a 3-bit number that describes the first time slot that a TDMA message will be sent in from a given station. Multiple ground stations may send the same SSID. The purpose of the SSID is to determine whether to parse more of the data message or ignore an incoming message without spending processing resources [1]. The Transmission Length segment is a 17-bit integer that describes how many bits are present in the application data and the Application FEC [1]. The Training Sequence FEC is a 5-bit block code that is generated from the SSID and Transmission Length segments. The SSID is concatenated with the Transmission Length before being multiplied by the transposition of parity matrix H [1]. Table 3.4 shows the value of parity matrix H. The following equation is used to calculate the Training Sequence FEC with least significant bits on the left of the matrices.

$$[P_{1...5}] = [SSID_{1...3}, TL_{1..17}]H^{T}$$

$$
H = \begin{array}{cccccccccccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1
\end{array}
$$

Table 3.4. ***Training Sequence FEC Parity Matrix****: Training FEC parity matrix H [1].*

The Application FEC is generated using a systematic fixed length Reed-Solomon (255,249) code. The Reed-Solomon code is generated in a Galois Field of size 2^8 with 8-bit symbols. The Application FEC is sent to Bit Scrambling most significant first [1]. Bit scrambling is performed on the message data using a pseudonoise scrambler with a 15 bit generator register. The initial value of the generator register is 100 1101 0100 1011 where the least significant bit is on the right. The bit scrambler is defined by the polynomial

$$X_{15} + X_1 + 1$$

where the subscripts represent the indices in the shift register [1]. The shift register is rotated by one from least significant to most significant each iteration. Addition is implemented as a single-bit exclusive-or. An exclusive-or is performed on each bit of the message data in a stream as the pseudonoise scrambler iterates. Figure 3.11 shows the pseudonoise scrambler algorithm.

3.3.4 VDB Message Development

The Telerad 9009 VDB transmitter uses a serial protocol to receive commands. LVBS 3.0 prepares the TDMA message into a formatted Telerad 9009 command

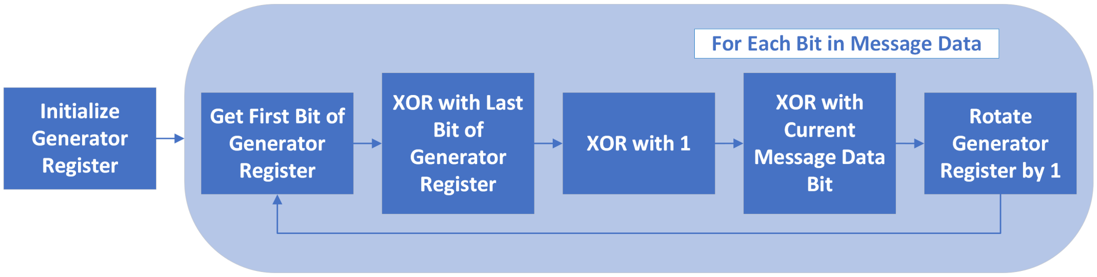

Initialize Generator Register
For Each Bit in Message Data
Get First Bit of Generator Register
XOR with Last Bit of Generator Register
XOR with 1
XOR with Current Message Data Bit
Rotate Generator Register by 1

Figure 3.11. ***Pseudonoise Scrambler Algorithm Diagram****The pseudonoise scrambler algorithm can be used to unscramble data that has been scrambled with the same method and generator register polynomial [1].*

message. All VDB transmitter messages include a Packet Sync segment, Message ID, Message Length, Application Data, and Checksum [28]. LVBS 3.0 only sends the TX CAT-I Message command to the transmitter during normal operation. After a message has been sent to the transmitter a Command Acknowledge message is returned to LVBS 3.0. In addition, the transmitter sends a health status message every 500 milliseconds [28]. Table 3.5 shows the contents of a TX CAT-I Message command in detail.

Table 3.5 VDB TX CAT-I Message Structure

Segment Name	Size (bytes)	Description
Packet Sync	2	Hexadecimal 00 FF
Message ID	1	Hexadecimal 5C
Message Length	1	Integer number of bytes in the application data segment
Application Data	225	Data contents for the TX CAT-I Message. Described in more detail in 3.6.

Checksum	2	A 16-bit additive checksum of the entire message except for the checksum.

Table 3.5. ***VDB TX CAT-I Message Structure****: Contents of a TX CAT-1 Message for a Telerad EM9009 VDB transmitter [28].*

Table 3.6 VDB TX CAT-I Application Data	
Size (bytes)	Description
1	Reserved Hexadecimal 00
1	Reserved Hexadecimal FF
1	Integer TDMA Transmission Slot
4 - 255	CAT-I LAAS Message

Table 3.6. ***VDB TX CAT-I Application Data****: Contents of a TX CAT-I Application Data for a Telerad EM9009 VDB transmitter [28].*

3.3.5 VDB Transmit Process

The prototype software never successfully established robust communication with the VDB transmitter. The VDB Transmit process developed in this work sends data that have been formed during the Type Message process to the VDB transmitter. The data enter the VDB Transmit process via a lossy single-element tag channel that is polled when the VDB Transmit process is ready to send a message, ensuring that only the latest data are sent and that timing remains structured. Acknowledgment and status messages from the VDB transmitter are received in the VDB Transmit process.

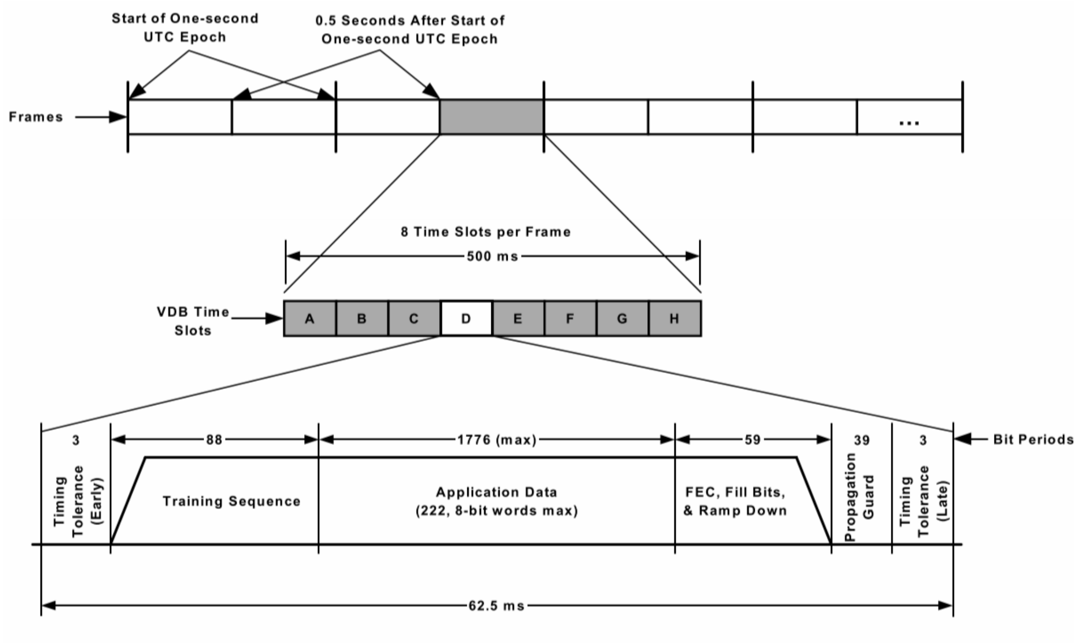


Figure 3.12. ***TDMA Timing Diagram****: Frames begin every every 500 milliseconds and are composed of eight time slots. A time slot may contain a single LAAS message [1].*

3.3.5.1 TDMA Communication Timing

TDMA communication is organized into frames that are 500 milliseconds long. Each frame is synchronized to UTC time. Each frame is divided equally into eight time slots that are 62.5 milliseconds long. The VDB transmitter must actively transmit in a time slot for five consecutive frames for the time slot to be used. Figure 3.12 shows a the timing structure used in TDMA communication from the RTCA DO-246E [1].

3.3.5.2 VDB Transmit Algorithm

The timed WHILE loop structure in LabVIEW synchronizes the start of the structure to a specified time, allows for the events inside the timed WHILE loop to use a defined timing source, and allows the iterations of the WHILE loop to occur at a set period. The VDB Transmit process begins by allocating serial communication resources. A timed WHILE loop is started that is synchronized to system time at the next one second interval. The timed WHILE loop uses a 1 MHz time source and iterates every 500 milliseconds. The timed WHILE loop has 8 subdivisions that are used to communicate with the VDB transmitter if messages are available and scheduled to be transmitted. The 8 subdivisions execute 62.5 milliseconds apart and correspond to two frames of TDMA time slots. The VDB transmitter sends health messages every 500 milliseconds and acknowledgment messages after each transmit command, that indicates if the message was transmitted successfully. The timed WHILE loop is synchronized and restarted if the response indicates the transmission was unsuccessful. Figure 3.13 shows the algorithm used for transmitting TDMA messages to the VDB transmitter.

3.3.5.3 VDB System Health Message

The system health message is sent from the VDB transmitter every 500 milliseconds. Table 3.7 shows the structure and contents of a system health message. Transmitter frequency is calculated by adding 100 MHz to the value provided in the health message [28]. Transmitter power is given in dBm and is calculated from the value provided in the health message according to equation 3.1 [28].

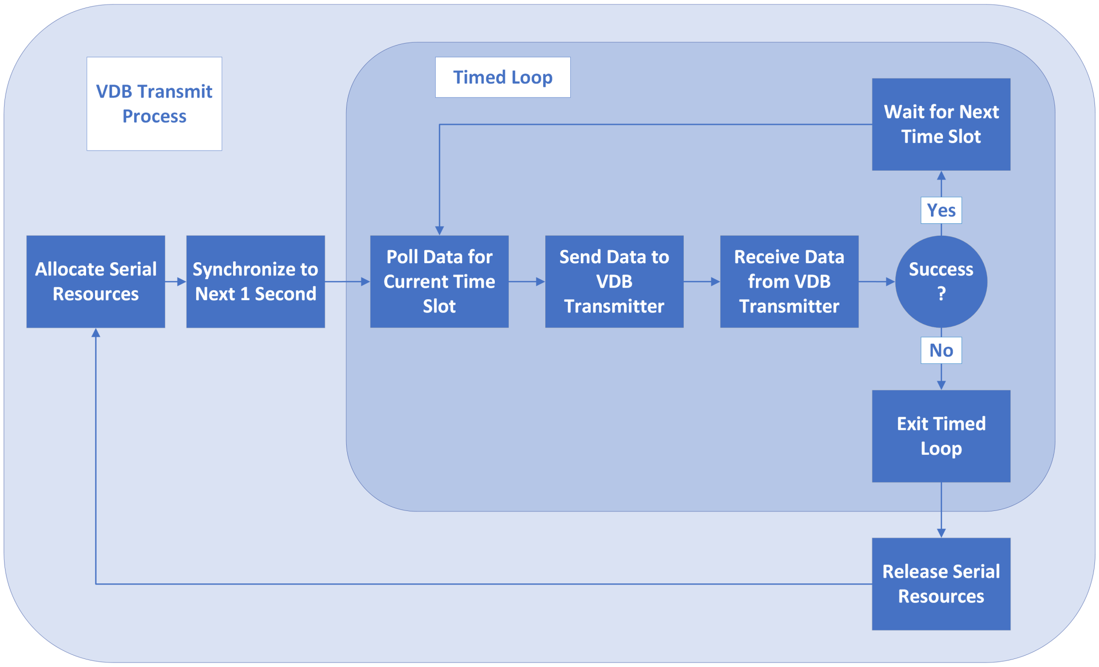

VDB Transmit Process
Timed Loop
Allocate Serial Resources
Synchronize to Next 1 Second
Poll Data for Current Time Slot
Send Data to VDB Transmitter
Receive Data from VDB Transmitter
Success ?
Yes
No
Wait for Next Time Slot
Exit Timed Loop
Release Serial Resources

Figure 3.13. ***VDB Transmit Algorithm Diagram**: The timed WHILE loop is started synchronized with the next upcoming second. The timed WHILE loop repeats the same process of sending a message and waiting for a response for each time slot that has data in it. The WHILE loop stops and synchronizes itself every time a message fails to be successfully acknowledged.*

$$F = P \cdot 10 - 300 \tag{3.1}$$

where

P = Output Power (dBm)

F = Integer Configuration Field for Output Power

Table 3.7 VDB Health Message

Segment Name	Size (bytes)	Description
Packet Sync	2	Hexadecimal 00 FF
Message ID	1	Hexadecimal 53
Message Length	1	Hexadecimal 08
Transmitter Failure	1	Failure: Hexadecimal 01 Normal: Hexadecimal 00
Transmit Frequency (Low Byte)	1	The lower byte of a 2 byte integer. Frequency is given in kHz.
Transmit Frequency (High Byte)	1	The lower byte of a 2 byte integer. Frequency is given in kHz.

Table 3.7 (*cont*)

TDMA Transmission Slots	1	Bit mask used to describe which time slots will be used to transmit messages. The first time slot is represented by the most significant bit and the last time slot is represented by the least significant bit. 1 indicates that a message will be sent. 0 indicates that a message will not be sent.
Output Power Setting	1	An integer value used to calculate power.
Detailed BIT Results	3	Detailed diagnostic data. Details are defined in table 3.8.
Checksum	2	A 16-bit additive checksum of the entire message except for the checksum.

Table 3.7. ***VDB Health Message**: Contents of an health message for a Telerad EM9009 VDB transmitter [28].*

Table 3.8 VDB Health BIT Results

Segment Name	Size (bits)	Description
Transmitter Failure	1	1 indicates failure.
Transmitter Disabled	1	1 indicates shut down.
Frequency Synthesizer out-of-lock	1	1 indicates error.

Table 3.8 (*cont*)		
Temperature too high	1	1 indicates error.
Supply too high	1	1 indicates VDC input is greater than 31 VDC.
Supply too low	1	1 indicates VDB input is lower than 24 VDC.
GPS 1PPS	1	1 indicates 1PPS signal is not present.
spare	1	not used.
EEPROM Error	1	1 indicates error.
Microprocessor RAM Error	1	1 indicates error.
Flash Memory Error	1	1 indicates error.
DSP RAM Error	1	1 indicates error.
DSP Error	1	1 indicates error.
spare	2	not used.
High VSWR	1	1 indicates error.
Output power measurement	8	Single byte integer that can be used to calculate the measured output power of the transmitter.

Table 3.8. ***VDB Health BIT Results***: *Detailed BIT contents from Telerad EM9009 VDB transmitter health message [28].*

3.3.5.4 VDB Acknowledgement Message

The command acknowledgment message is sent within 20 milliseconds of the VDB transmitter receiving a command. Table 3.9 describes the structure of the message.

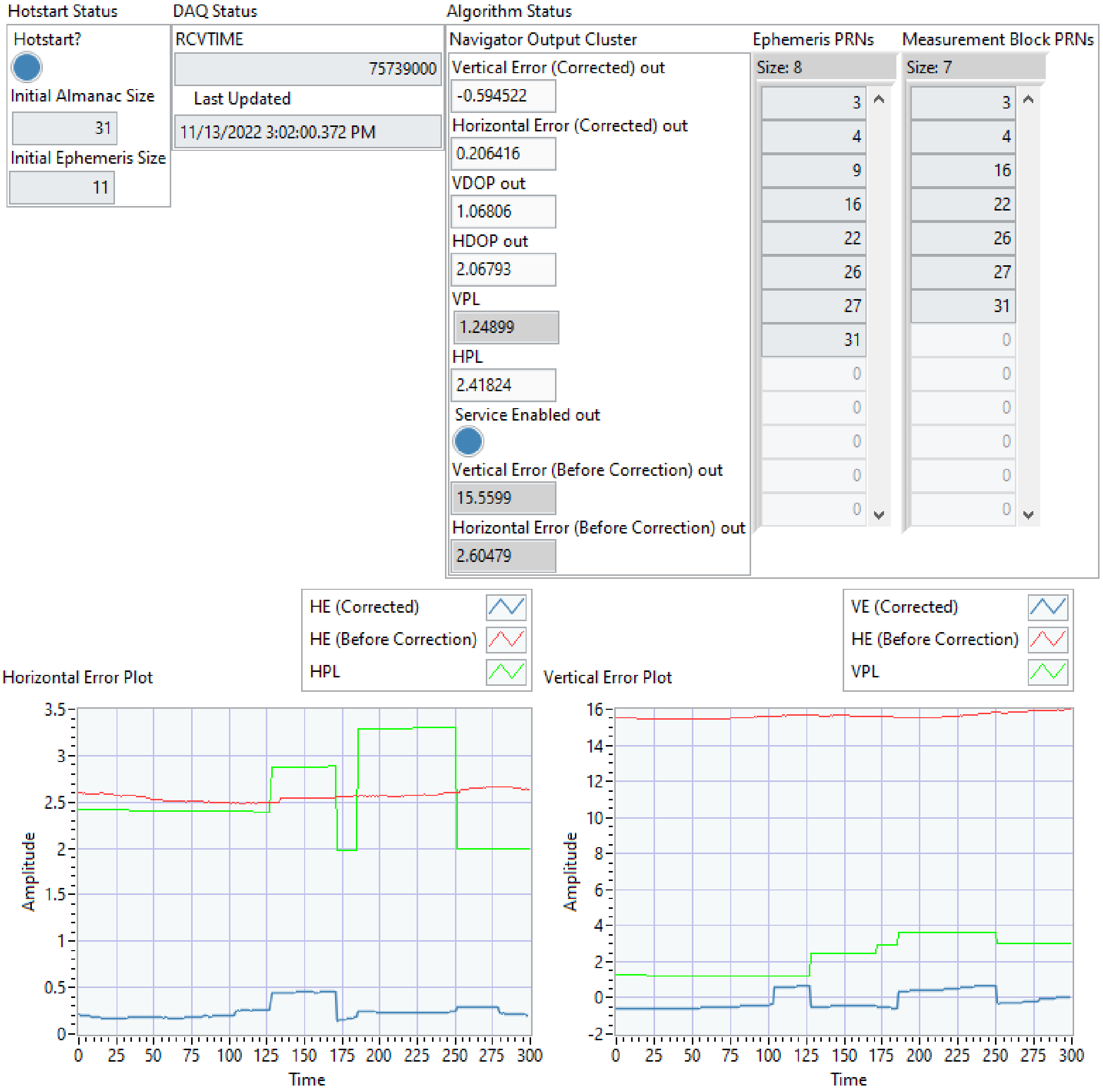
Table 3.9 VDB Acknowledgement Message		
Segment Name	Size (bytes)	Description
Packet Sync	2	Hexadecimal 00 FF
Message ID	1	Hexadecimal 96
Message Length	1	Hexadecimal 01
Command Status	1	Successful: Hexadecimal 01 Failure: Hexadecimal 00
Checksum	2	A 16-bit additive checksum of the entire message except for the checksum.

Table 3.9. ***VDB Acknowledgement Message****: Detailed BIT contents from Telerad EM9009 VDB transmitter health message [28].*

Chapter 4

Results

The LVBS 3.0 software developed in this thesis functions in a real time capacity and satisfies the GBAS timing requirements. Figure 4.1 shows an image of the LVBS 3.0 front panel user interface while the program is running live. The LVBS 3.0 user interface displays the output status from the hotstart script, which provides the user information about the initial almanac and ephemeris that is useful for determining the health of the system at startup. The time of last data acquisition, in both GPS time and in system time, is available on the UI to provide the user information about the status of the DAQ process. Plots and numeric indicators display the vertical error (VE), horizontal error (HE), vertical protection level (VPL), horizontal protection level (HPL), vertical dilution of precision (VDOP), and horizontal dilution of precision (HDOP) that are generated using the CL-GBAS algorithms. These provide an indication of the quality of the corrections being gneerated in the Type 1 message from the Algorithm process. The current set of SVs present in the ephemeris data in the Algorithm process are visible in the Ephemeris PRNs list on the UI. This provides an indication of how many SVs may be potentially used in the solution. The Measurement Block PRNs list displays the current set of SVs for which corrections are being generated to populate the LAAS Type 1 measurement block.



Hotstart Status

Hotstart? ●

Initial Almanac Size 31

Initial Ephemeris Size 11

DAQ Status

RCVTIME 75739000

Last Updated 11/13/2022 3:02:00.372 PM

Algorithm Status

Navigator Output Cluster

Vertical Error (Corrected) out -0.594522

Horizontal Error (Corrected) out 0.206416

VDOP out 1.06806

HDOP out 2.06793

VPL 1.24899

HPL 2.41824

Service Enabled out ●

Vertical Error (Before Correction) out 15.5599

Horizontal Error (Before Correction) out 2.60479

Ephemeris PRNs	Measurement Block PRNs
Size: 8	Size: 7
3	3
4	4
9	16
16	22
22	26
26	27
27	31
31	0
0	0
0	0
0	0
0	0
0	0

Figure 4.1. ***Software Front Panel**: An image of the front panel of the status of LVBS 3.0 while it is running.*

The plot on the lower left side of figure 4.1 shows the HE before and after LAAS corrections are applied, as well as the HPL. The plot on the right side of figure 4.1 shows the VE before and after LAAS corrections are applied, as well as the VPL.

The Minimum Aviation System Performance Standards for LAAS (MASPS) evaluates LAAS based on accuracy, integrity and continuity and classifies them according to GBAS Service Levels (GSLs). The functionality of the LVBS 3.0

software that was developed in this thesis was verified using the short term bench test, simulation, and analysis parameters defined in MASPS. The LVBS 3.0 system was tested to verify that the SAD measurements meet the accuracy requirements defined in FAA-2937.

4.1 Navigational System Error (NSE) Analysis

The accuracy of the LVBS 3.0 system was evaluated by applying corrections to the position acquired from a receiver with a known surveyed position, and comparing the VE and HE before and after correction. Figure 4.2 compares the VE generated before corrections are applied to the system and the VE generated after the corrections from the Type 1 measurement block are applied to the system. The VE makes up the largest proportion GPS solution error and this is seen in figures 4.2 and 4.4. The uncorrected VE is greater during daylight hours be ionosphere is more active in the sun.

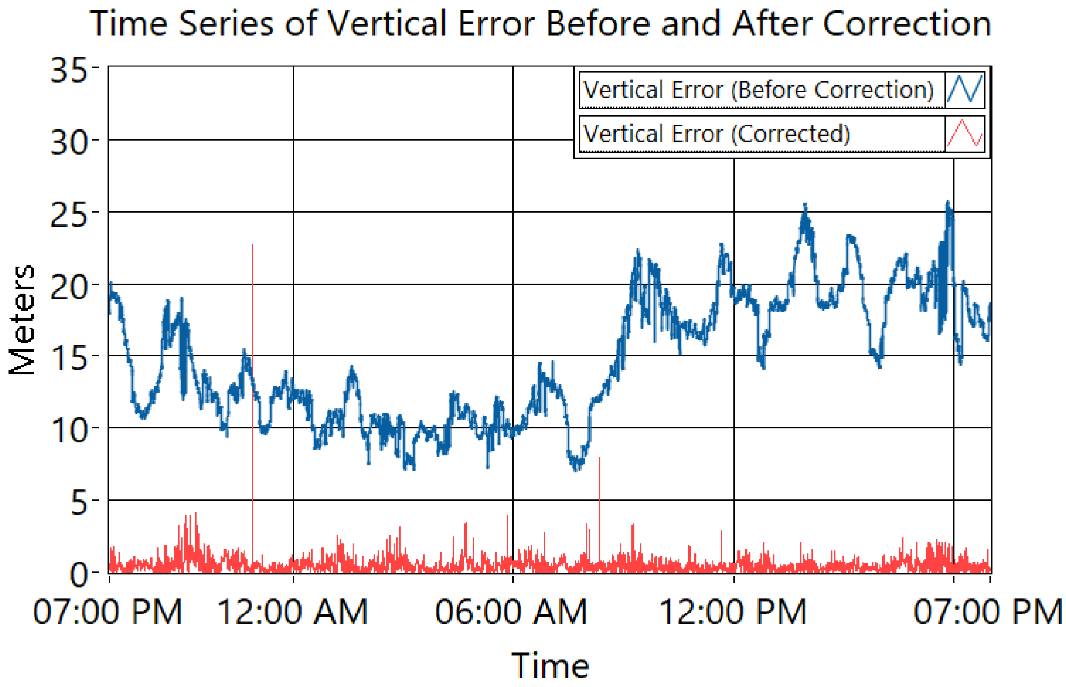


Figure 4.2. ***Vertical Error Time Series Plot:*** *The blue plot represents the VE that is calculated from the position directly from the GPS receiver. The red plot represents the VE after the Type 1 corrections are applied to the system.*

Figure 4.2 contains a spike where the VE after corrections exits the acceptable range of error for GSL F. This spike in VE is caused by a rapid loss of availability from most SVs in the Type 1 LAAS message. Shortly after the spike, the service becomes unavailable for eight seconds before returning to normal operation. Figure 4.3 shows the VE overlaid with the number of SVs used in the solution. The primary benefit of the LM concept from CL-GBAS implemented in this thesis enables the measurement of VE in real time before corrections are transmitted. Future work should include utilizing the feedback, provided by the LM, to maintain only high quality corrections and remove HMI more completely. Continuity is described in more detail in Section 4.2.

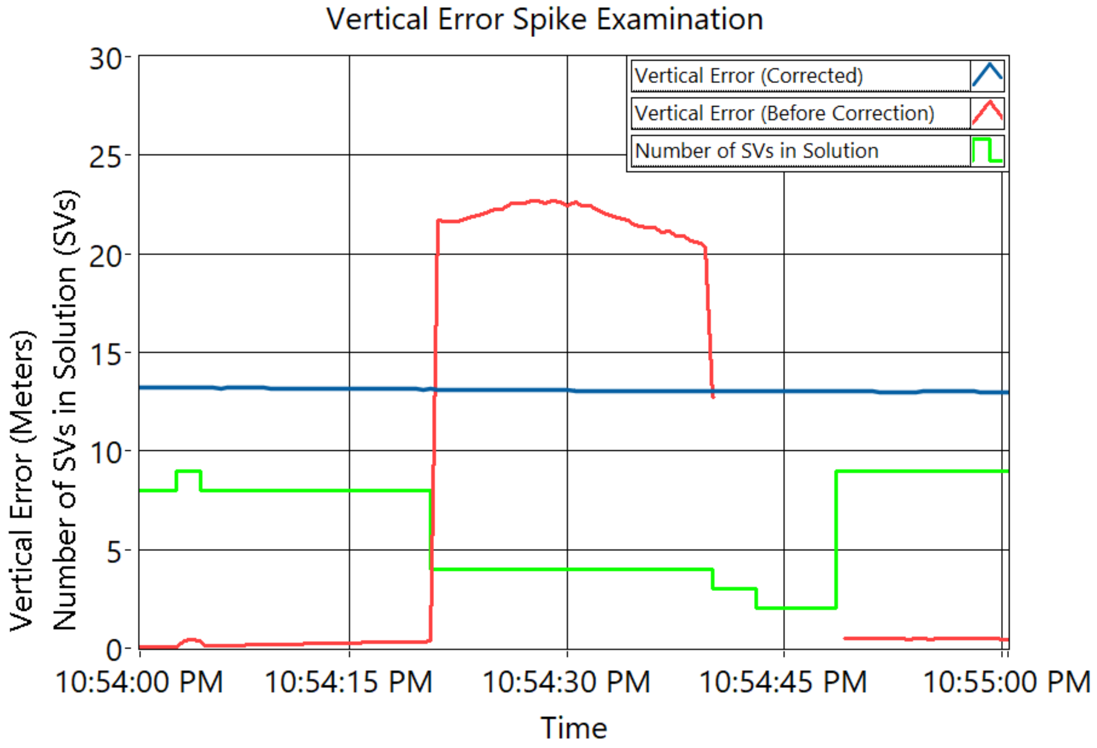


Figure 4.3. ***Vertical Error Spike Examination****: The abrupt increase in VE is caused by rapidly losing available SVs in the GBAS solution. The green curve shows the number of available SVs present in the GBAS correction. After more SVs become available to the solution, the corrected VE returns to normal levels. The ability to monitor the quality of the GBAS corrections in real time is possible with the presence of LM from the CL-GBAS concept.*

The HE generated before and after corrections are applied are presented in figure 4.4. The uncorrected HE is smaller than the uncorrected VE shown in 4.2. The MASPS classifies GSL according to horizontal (referred to in MASPS as lateral accuracy) and vertical accuracy. Table 4.1 describes the accuracy classifications for LAAS. The accuracy assumes a normal distribution of error that is zero mean and has two standard deviations as indicated in table 4.1 (e.g. for GSL A, $\mathcal{N}(0,8)$).

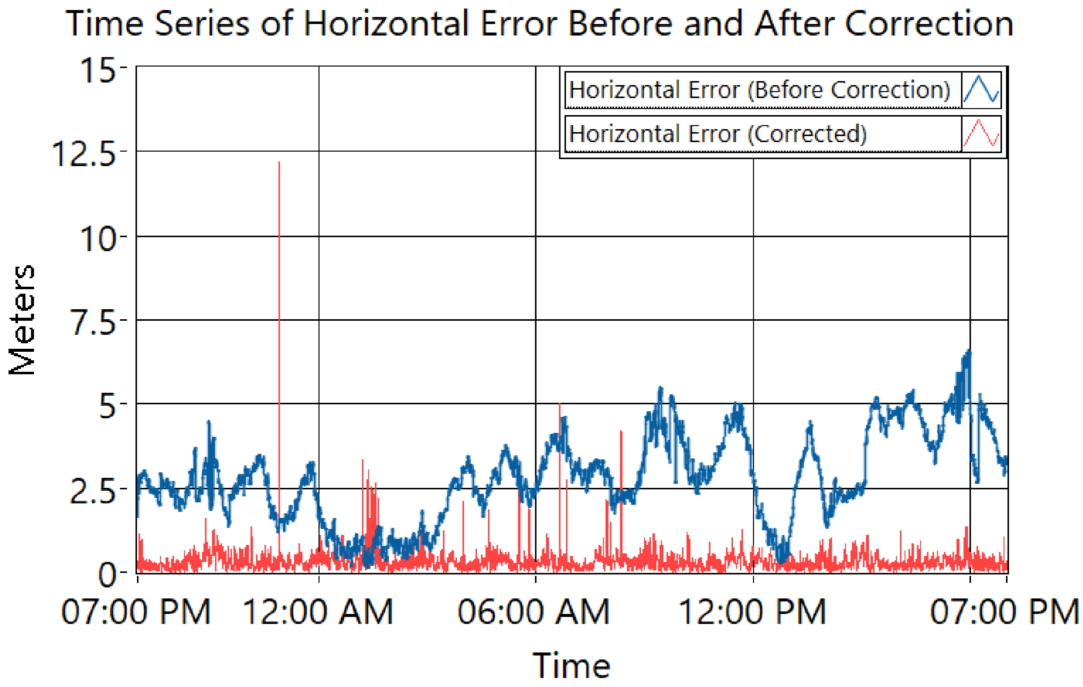


Figure 4.4. ***Horizontal Error Time Series Plot***: *The blue plot represents the HE that is calculated from the position directly from the GPS receiver. The red plot represents the HE after the Type 1 corrections are applied to the system.*

Table 4.1 NSE Standards from MASPS		
GBAS Service Level	Lateral Accuracy NSE 95% (meters)	Vertical Accuracy NSE 95% (meters)
A	16.0	20
B	16.0	8.0
C	16.0	4.0
D, E, F	5.0	2.9

Table 4.1. ***NSE Standards from MASPS***: *These accuracy performance standards are provided by the MASPS for LAAS [29].*

The NSE data were collected over 24 hours while performing a bench test using the surveyed LM as the navigator. Therefore, no actual flight was required to verify the VE and HE performance. The VE and HE are generated using the surveyed LM antenna. The NSE data were sorted into a histogram and the bin frequency values were adjusted by the sample size to represent probability distribution values. These probability distribution values were overlaid with curves that represent the lateral and vertical accuracy distributions described in Table 4.1. Figures 4.5 and 4.7 show the VE and HE, respectively, that are computed from ranging data collected directly from the GPS receiver without applying the LAAS algorithm. Figures 4.6 and 4.8 show the VE and HE, respectively, that are calculated from the position that is generated after applying the corrections from the Type 1 messages to the same receiver data shown in figures 4.5 and 4.7. It should be made clear that both NSE data sets are generated from the same set of measurements from the GPS receiver.

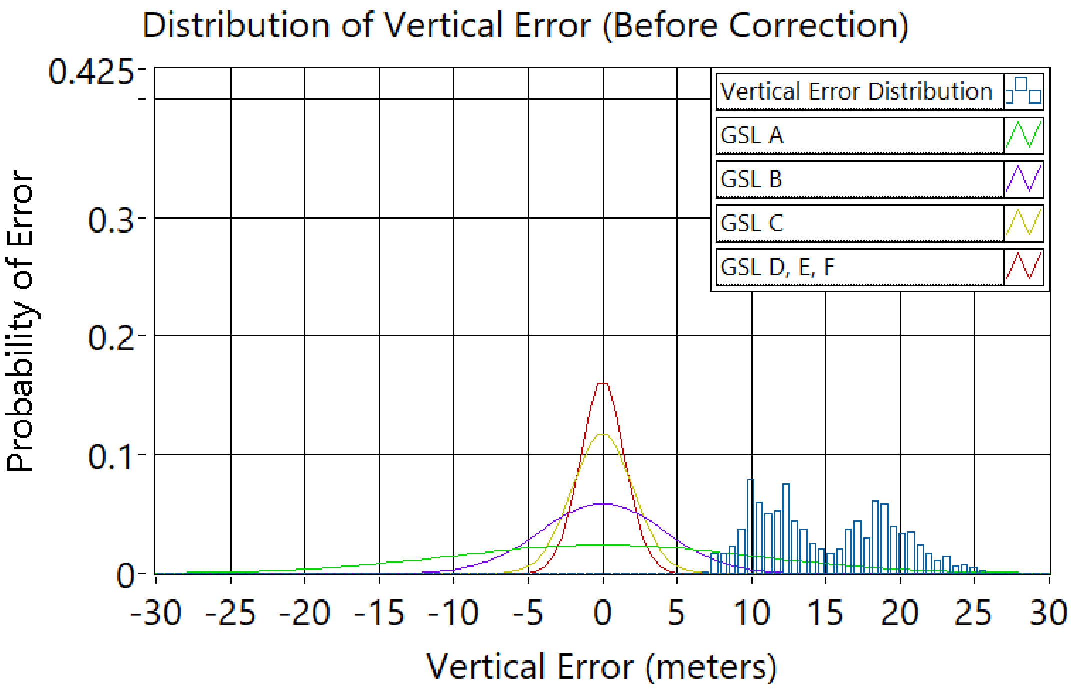


Figure 4.5. ***Vertical Error Probability Distribution Before Correction****: The distribution of vertical error in meters before the LAAS correction algorithm is applied to the receiver position solution.*

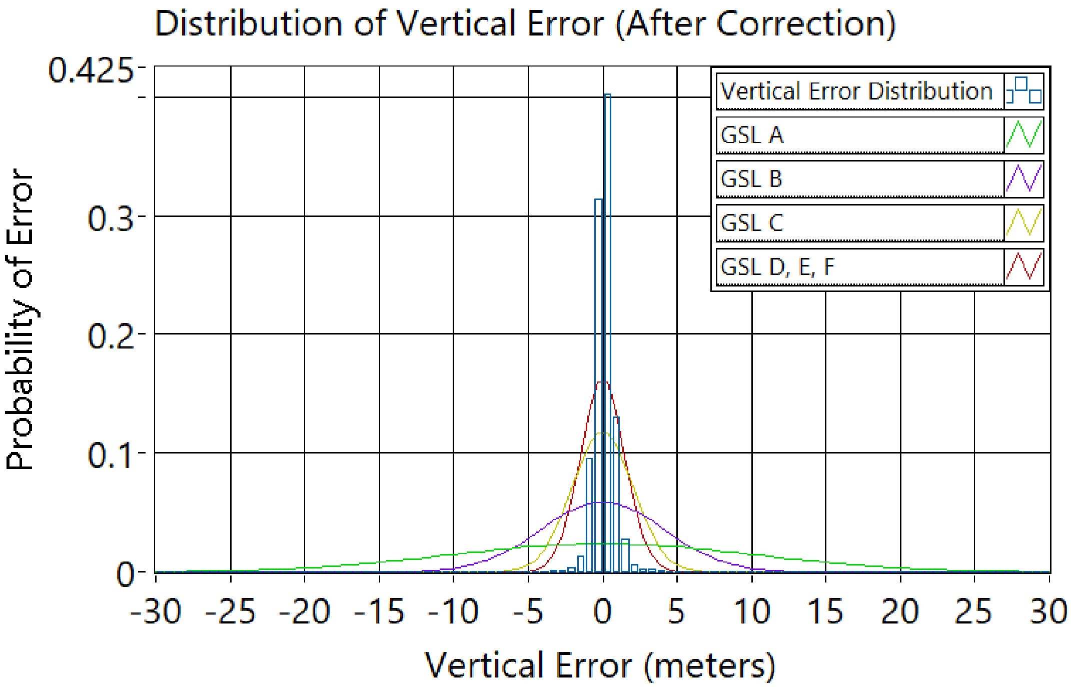


Figure 4.6. ***Vertical Error Probability Distribution After Correction****: The distribution of vertical error in meters after the LAAS correction algorithm is applied to the receiver position solution.*

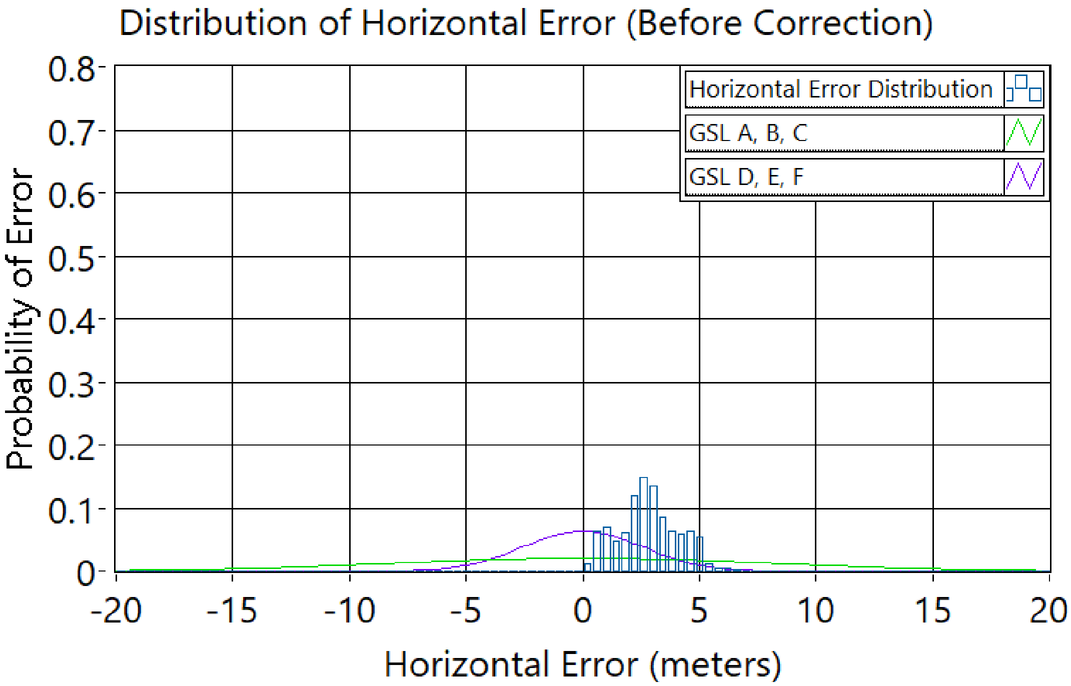


Figure 4.7. ***Horizontal Error Probability Distribution Before Correction****: The distribution of horizontal error in meters before the LAAS correction algorithm is applied to the receiver position solution.*

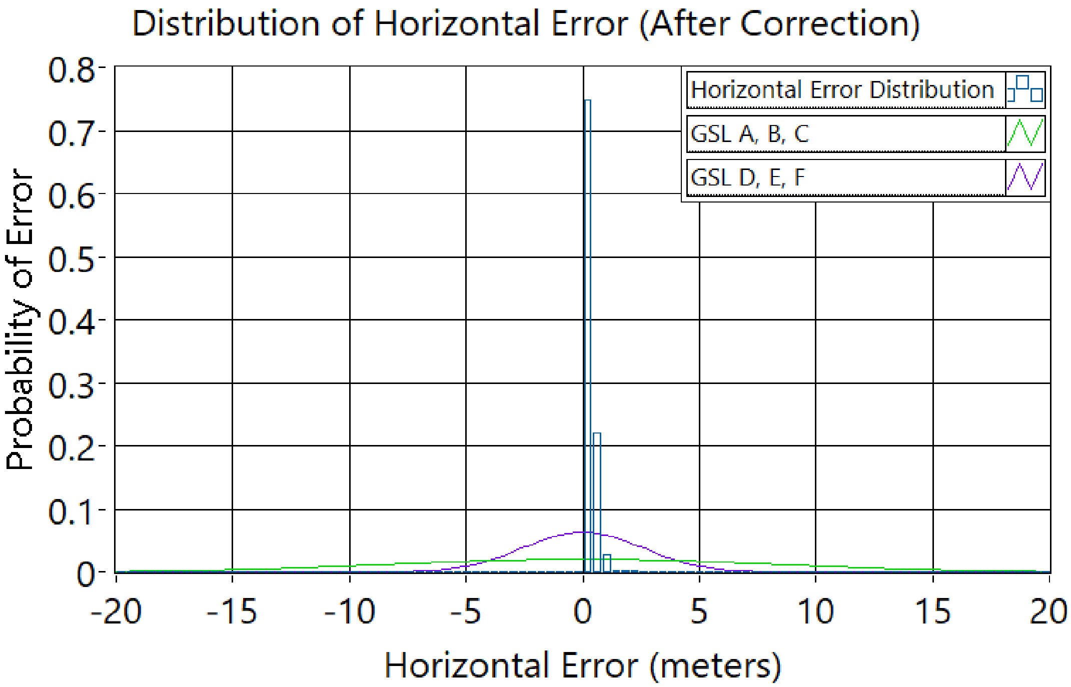


Figure 4.8. ***Horizontal Error Probability Distribution After Correction***: *The distribution of horizontal error in meters after the LAAS correction algorithm is applied to the receiver position solution.*

Before correction, the vertical and horizontal accuracy of the GPS receiver are distributed outside of the acceptable range for any GSL. After the LAAS corrections are applied to the position, the 95% vertical accuracy is 1.43425 meters and the 95% lateral accuracy is 0.574413 meters. The results of the bench tests indicate that the accuracy corrections that are generated by the software developed in this thesis satisfy the highest level of GSL (F). Table 4.2 summarizes the results of the accuracy test.

Table 4.2 Accuracy Test Results							
Before Correction				After Correction			
$\overline{\text{HE}}$	$2\sigma_{\text{HE}}$	$\overline{\text{VE}}$	$2\sigma_{\text{VE}}$	$\overline{\text{HE}}$	$2\sigma_{\text{HE}}$	$\overline{\text{VE}}$	$2\sigma_{\text{VE}}$
2.8042	2.59306	14.8494	8.78306	0.31561	0.58614	0.09289	1.46352

Table 4.2. ***Accuracy Test Results***: *Accuracy test results before and after corrections for vertical and horizontal error. All units are meters.*

4.2 Continuity Analysis

The vertical protection level (VPL) and horizontal protection level (HPL) are used to evaluate the risk that the VE and HE are unbounded. The method used to calculate the VE, HE, VPL and HPL are available in section D.2.7. The integrity of a LAAS system is the probability that the system will send an alert message within an acceptable time frame after it has been generated. The continuity of a LAAS system is the probability that a LAAS system will provide correction data. This thesis focuses on the probability that the protection levels do not violate the alarm levels. The vertical alarm level (VAL) and lateral alarm level (LAL) are used to evaluate when it is appropriate to generate an alert. Table 4.3 describes the continuity and integrity classifications for LAAS. The service levels in the left most column of Table 4.3 are the same service levels referred to in table 4.1.

Table 4.3 Integrity and Continuity Requirements

GSL	Integrity Probability	Time to Alert (seconds)	LAL (meters)	VAL (meters)	Continuity Probability
A	$1 - 2 \times 10^{-7}$ in any 150s	10	40	50	$1 - 8 \times 10^{-6}$ in any 15s
B	$1 - 2 \times 10^{-7}$ in any 150s	6	40	20	$1 - 8 \times 10^{-6}$ in any 15s
C	$1 - 2 \times 10^{-7}$ in any 150s	6	40	10	$1 - 8 \times 10^{-6}$ in any 15s
D	$1 - 2 \times 10^{-7}$ in any 15s vert, in any 30s lat	2	17	10	$1 - 8 \times 10^{-6}$ in any 15s

Table 4.3 (*cont*)					
E	$1 - 2 \times 10^{-7}$ in any 15s vert, in any 30s lat	2	17	10	$1 - 4 \times 10^{-6}$ in any 15s
F	$1 - 2 \times 10^{-7}$ in any 15s vert, in any 30s lat	2	17	10	$1 - 2 \times 10^{-6}$ in any 15s vert, in any 30s lat

Table 4.3. ***Integrity and Continuity Requirements****: These integrity and continuity performance standards are provided by the MASPS for LAAS [29].*

Continuity data for this analysis were collected over 24 hours. Histograms were generated from the VPL and HPL data collected for this analysis. The bin frequency values of each histogram were scaled into probability distributions for each data set. Figures 4.9 and 4.10 present the probability distributions of the VPL and HPL, respectively, from the collected data. VPL and HPL were calculated using the K_{ffmd} specified for GSLs A, B, and C, as well as GSLs D, E, and F. The distributions were overlaid with vertical lines representing the VAL and LAL thresholds for the different GSLs. During the continuity test the Type 1 messages contained more than three measurement block elements (the minimum to be considered valid) 92.0547% of the time. The continuity of the OU LAAS can be improved by implementing an additional RS so that the LM will exist with an independent receiver and antenna, changing the locations of the existing RS to more ideal locations for multipath mitigation, and lowering the elevation masks on the GG12 GPS receivers so that the LVBS 3.0 software developed in this thesis may begin validating the SVs before are above 5° of elevation over the horizon. The LVBS 3.0 software developed in this thesis provides the environment necessary to easily develop algorithm enhancements to improve the continuity metrics and a straightforward way to test hardware modifications. Table 4.4 contains the probabilities of violating continuity over the different conditions defined for the different GSLs in table 4.3 when enough Type 1 corrections are available.

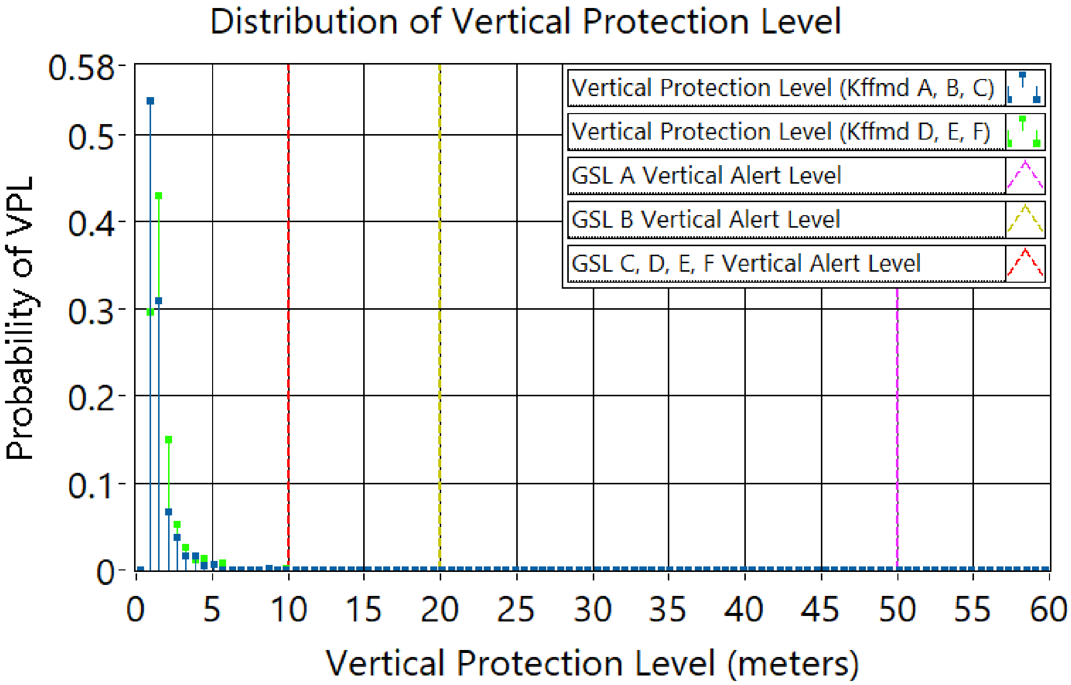


Figure 4.9. ***Vertical Protection Level Probability Distribution****: The distribution of vertical protection level calculated using the K_{ffmd} required for GSLs A, B, C and GSLs D, E, F respectively. The dashed vertical lines represent the alarm limits.*

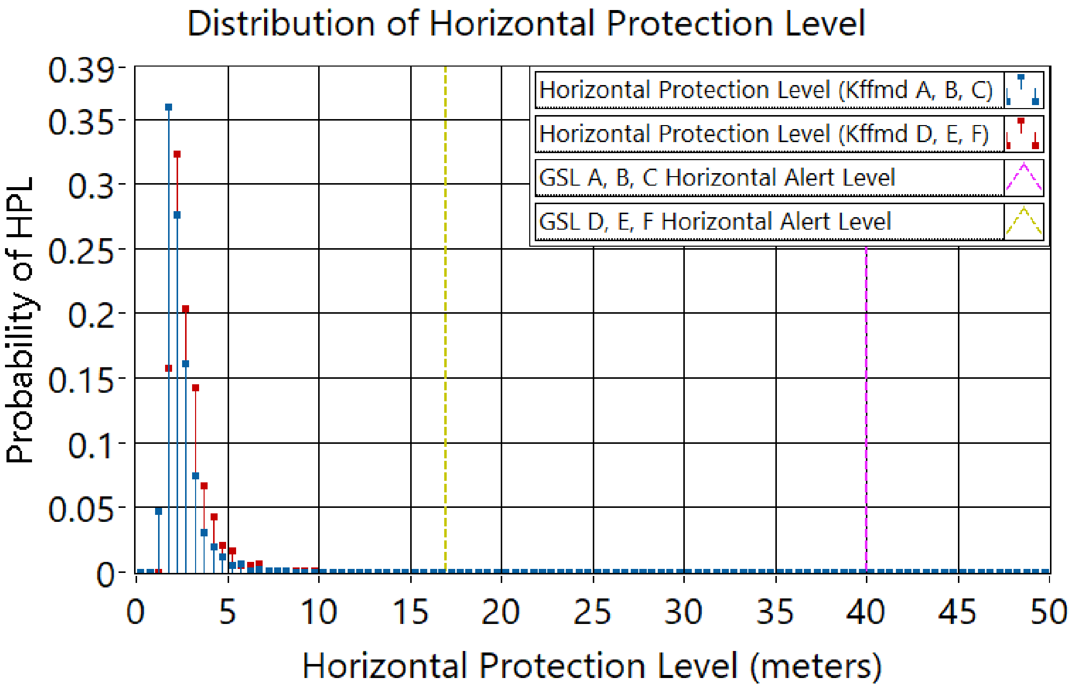


Figure 4.10. ***Horizontal Protection Level Probability Distribution****: The distribution of horizontal protection level calculated using the κ_{ffmd} required for GSLs A, B, C and GSLs D, E, F respectively. The dashed vertical lines represent the alarm limits.*

Table 4.4 Continuity Probability Results from Protection Level Violations	
GSL	Continuity Probability
A	$1 - 2.78284 \times 10^{-8}$
B	$1 - 2.78284 \times 10^{-8}$
C	$1 - 3.03194 \times 10^{-7}$
D	$1 - 8.76598 \times 10^{-7}$
E	$1 - 8.76598 \times 10^{-7}$
F	$1 - 9.38895 \times 10^{-7}$

Table 4.4. ***Continuity Probability Results from Protection Level Violations****: The data in this table was generated from the data protection level data collected over 24 hours.*

4.3 Source Availability Duration Analysis

The Source Availability Duration (SAD) was analyzed by comparing the SAD calculated in the algorithm process to the actual SAD. The SAD is calculated the entire time that an SV is available (i.e. in-view) and the maximum value that the SAD algorithm generates is three hours. A typical SV pass, over a given user position, can be as long as six hours, depending on the orbit. Figure 4.11 shows the elevation of an SV being used by LVBS 3.0. The availabiity mask is set to 5° so the elevation measure is truncated at the 5° mark when the satellite sets on the horizon. However, due to the periodic nature of the transmitted ephemeris information, the satellite does not become available for use on rising above the horizon until about 7.5° elevation. It is important to recognize that, when the satellite rises, it cannot be used as a ranging source for the LAAS until it has a valid ephemeris, so the availability on rising generally occurs sometime after the satellite has become visible on the horizon. Figure 4.12 shows the SAD as calculated by the algorithm process, developed in this work, for the same period.

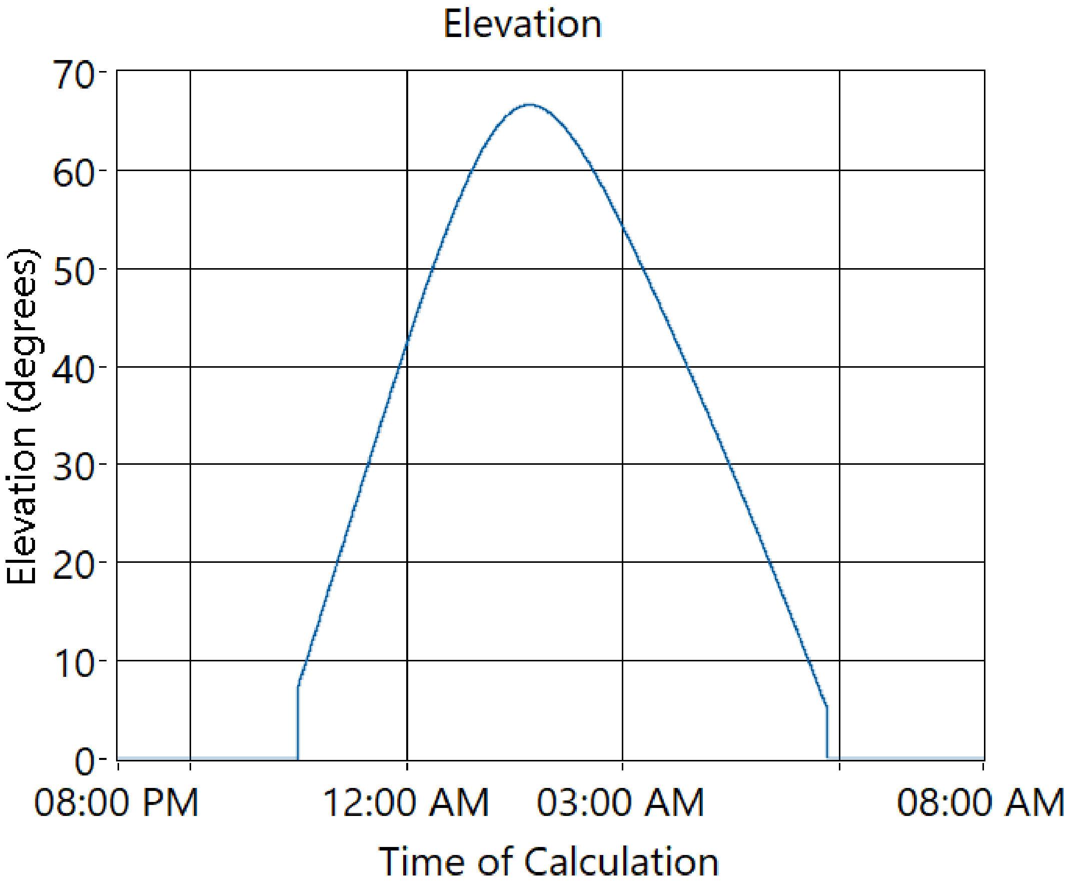


Figure 4.11. ***Space Vehicle Elevation Plot***: *The elevation of an SV being used by LVBS 3.0. The SV is no longer in use when the elevation falls below 5°. The SV became available shortly after it was 5° over the horizon because the GG12 receivers only output ephemeris information every 15 minutes and the GG12 elevation mask was set to 5° over the horizon. After the ephemeris data were received for this SV from the GPS receiver, there is a required two-minute validation period before the SV can be used. The elevation mask can be set to a lower value to allow the SVs to become available for use at lower elevations.*

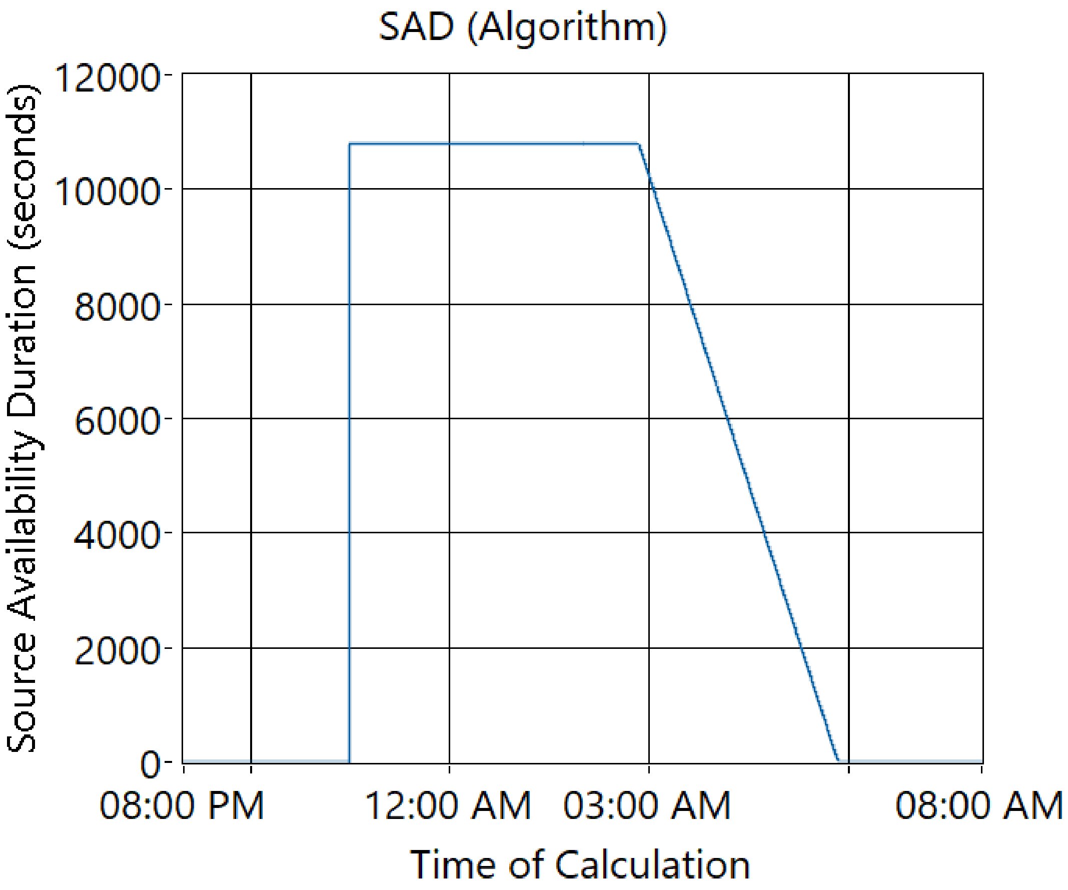


Figure 4.12. ***Source Availability Duration Generated Real-Time***: *This plot shows the SAD calculated during the standard operation of LVBS 3.0 developed for this thesis. The flat portion during the first four hours of availability is due to the maximum SAD value. Once the availability falls below the three-hour mark, the plot begins to drop linearly until the SV becomes unavailable at the 5° elevation point. Details on the method used to calculate this are available in section D.1.7.*

The SAD algorithm was tested by computing the actual SAD from collected data during post-processing. When the true SAD determined from the collected data exceeded three hours, it was truncated to three hours so that a comparison could be made with the real-ime SAD computation in this work. Figure 4.13 shows the actual SAD that is used for comparison.

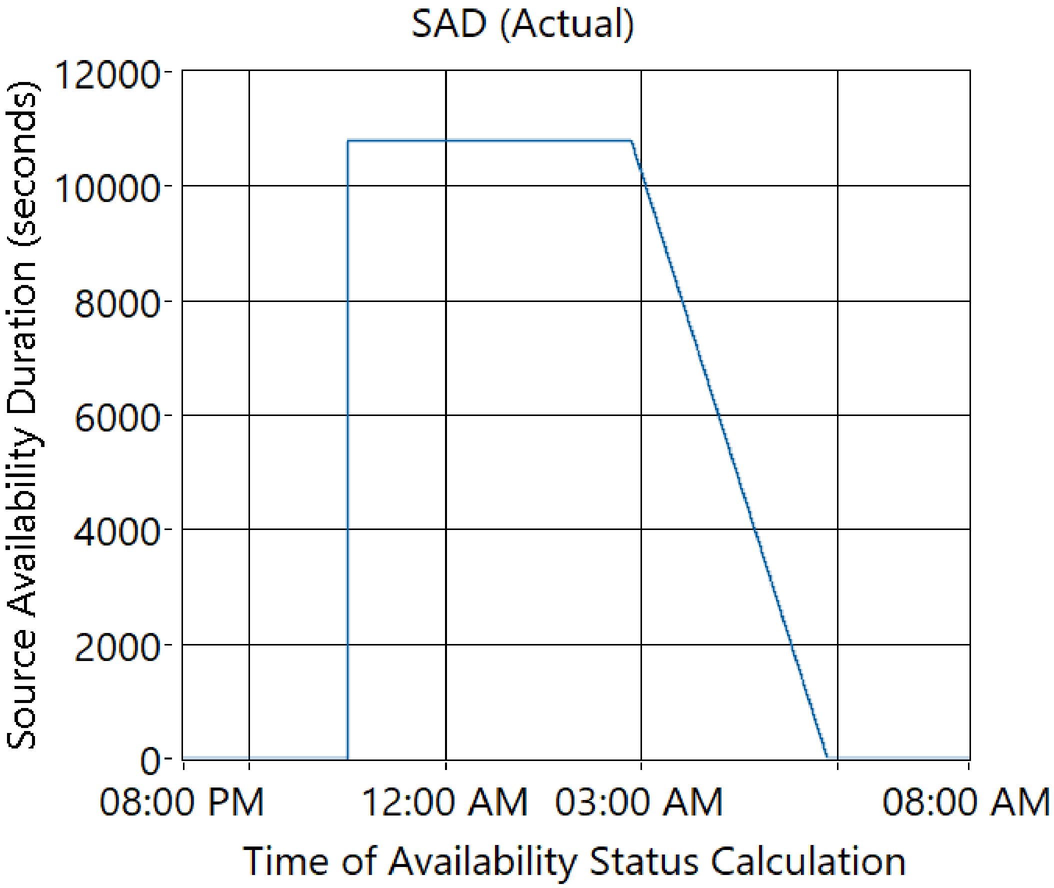


Figure 4.13. ***Source Availability Duration Generated From Recording***: *The SAD calculated based on the recorded availability of the SV. The SAD is bounded between 0 seconds and 10800 seconds (3 hours) and is 0 when the SV is unavailable.*

The SAD error is measured in seconds, where the difference represents the error between the true SAD (from post-processing) and the computed SAD in real time. The SAD error is calculated by subtracting the algorithm SAD value from the actual, post-processed SAD value during the time where the SAD is changing (i.e. from about 3:00AM until just before 6:00AM in the SAD figures). Figure 4.14 shows the probability distribution of the SAD error. The mean of the SAD error is 0.199122 seconds and the variance of the SAD error is 0.0751952 seconds (+/- 0.2742 standard deviation) from this evaluation.

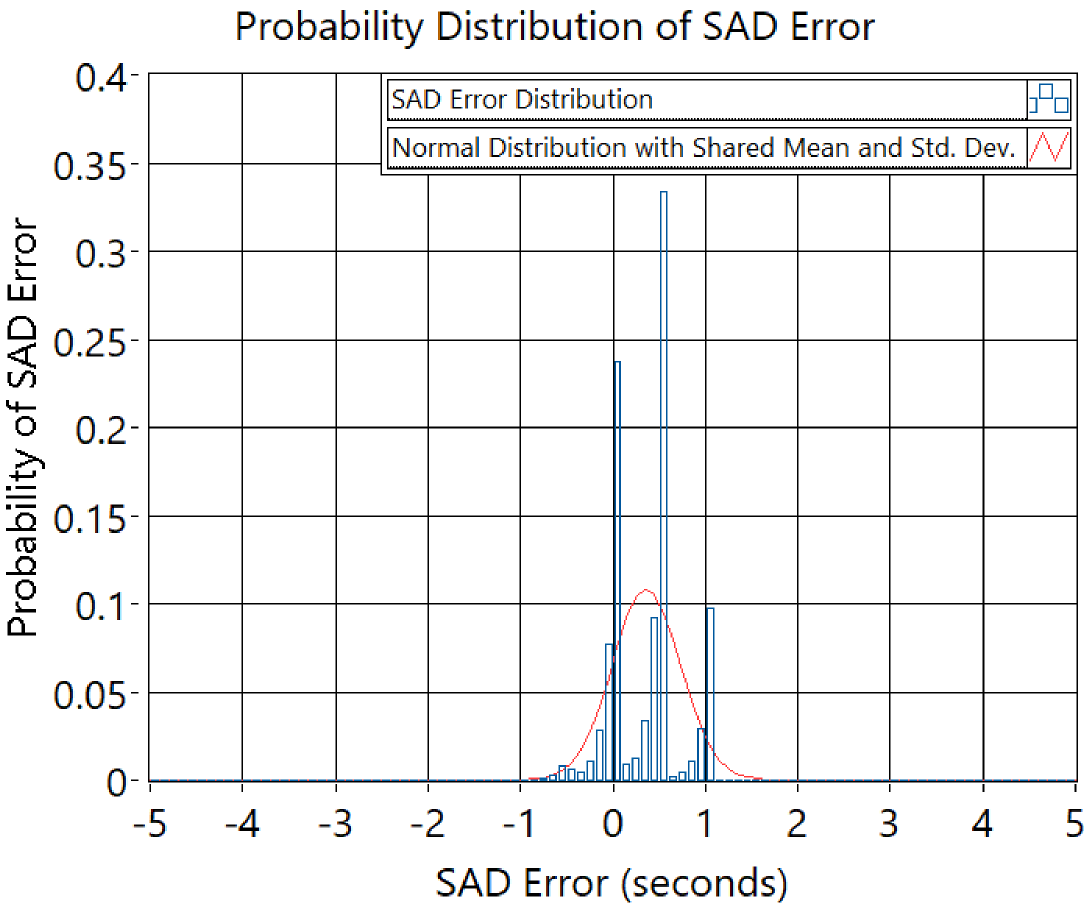


Figure 4.14. ***Source Availability Duration Probability Distribution**: The SAD error is calculated for the region of data where the SAD is less than 10800 seconds and greater than 0 seconds while the SV is available. The SAD error distribution is normalized.*

The SAD error was computed for data sets collected over 21 test cases using this methodology. Figure 4.15 shows the combined data error accumulated over multiple test cases.

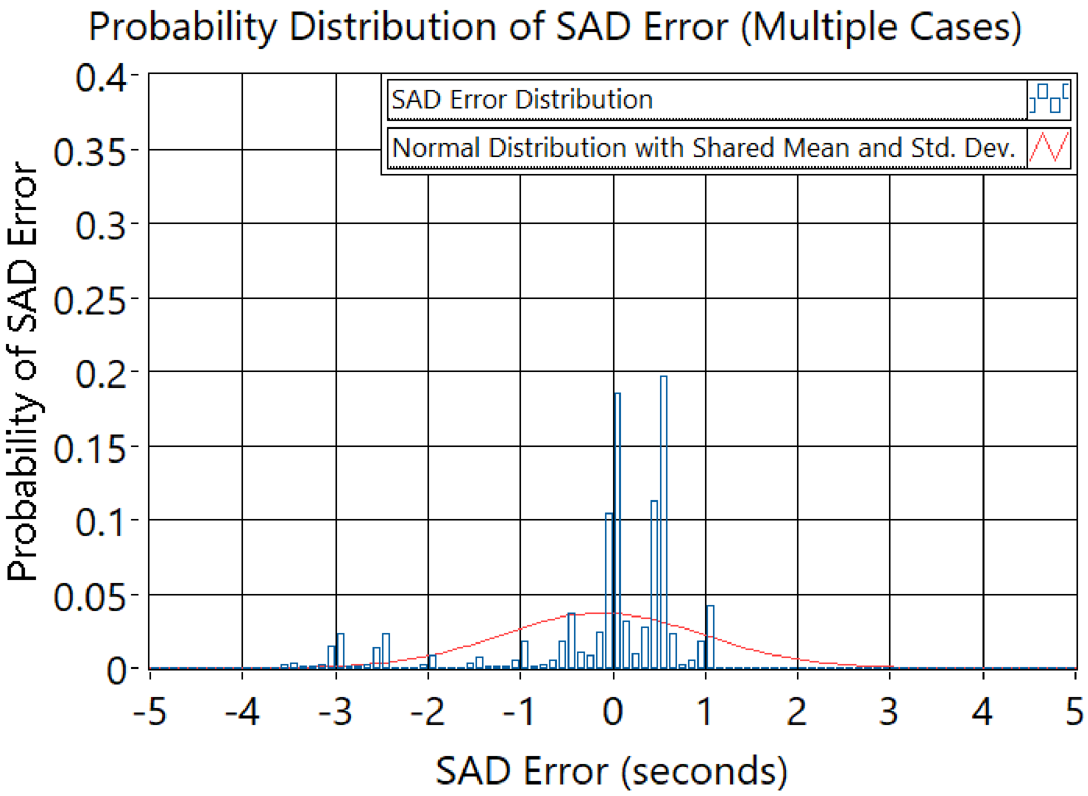


Figure 4.15. ***Source Availability Duration Probability Distribution From Multiple Tests****: The SAD error data from 21 test cases is combined to form this distribution.*

Using a larger data set it is evident that there is a broader distribution of error biased toward the real-time algorithm over-estimating the availability duration by as much as 3 seconds. In this case the distribution has a mean of -0.112246 seconds and a standard deviation of 1.08265 seconds. Although this error continues to meet the FAA specification, an improvement to this error can be realized by improving the Reference Station locations. The work done in this thesis provides a software structure and environment that facilitates testing and analyzing modifications of this type.

4.4 Summary of Results

The fundamental purpose of the LAAS is to reduce a user's navigational error during the critical phases of flight. This is encapsulated in the concept of the Navigational System Error (NSE). Assessing the performance of the LVBS 3.0 software architecture developed in this thesis resolves to analyzing user NSE before and after corrections generated by the LVBS 3.0 software. A test flight was not practical for this work, but the Local Monitor (LM) was well-suited to providing the navigator perspective. The accurately-surveyed LM coordinates are known, so determining error was a straightforward matter of supplying corrections to the LM, computing its position before and after corrections, and comparing both position solutions to the known, true position. The NSE analysis shows that the accuracy of the corrections that are generated by LVBS 3.0 are accurate enough to satisfy GSL F. As expected, the raw vertical error before corrections had a larger bias than the horizontal error. However, the corrections generated by the LVBS 3.0 reduced the vertical error from a mean of about 15 meters to a mean of 0 meters +/- 1.4 meters. Mean horizontal error before corrections was about 3 meters and was reduced to 0 meters +/- 0.6 meters. Optimum RS placement could improve these results even further, though RS placement is constrained in the research environment due to airport regulation. The continuity of LVBS 3.0 reached 92%, which needs to be increased to satisfy the GSL standards. However, the reduction in continuity was almost solely due to periods when fewer than four SVs were available in the corrections block, causing the system to not provide corrections during that time. This is also likely due non-optimum RS placement causing restricted views of the entire in-view constellation and excessive multipath for lower-elevation satellites. The

SAD Algorithm developed in this thesis satisfies the requirement for the predicted SAD to be accurate within 60 seconds of the actual SAD.

The analysis of NSE in this chapter clearly demonstrates that the work done in this thesis to determine an appropriate software architecture, and develop proper software techniques within the LabVIEW programming environment, has successfully produced an operational GBAS that can be used to thoroughly research areas of interest in system development from both the hardware and algorithm perspectives. The pipelining and parallelism techniques ensure that the real-time system is operational and provides corrections in a timely manner over many days. The modularity of the code and the class structure enhance the ability to assess multiple GPS receivers, and even potentially use different GPS receivers within the same LAAS to produce valid correction data.

Chapter 5

Conclusion

This thesis describes the application of the CL-GBAS algorithm to a real time Ground Based Augmentation System (GBAS) developed in LabVIEW. The results produced a viable system architecture and developed robust software in the LabVIEW environment. Using the LVBS 3.0 software developed in this work, the effectiveness of the CL-GBAS algorithm was demonstrated when applying and computed corrections to the Local Monitor. This work also demonstrates that the LVBS 3.0 accuracy qualifies as GSL F.

5.1 Contributions

A central contribution of this work is the development of a software architecture that ensures consistent operation, while also providing the flexibility to alter parts of the system for future research and easily analyze those alterations. The fundamental choice of pipelining and parallelism for the software developed in this thesis was essential to meeting the operational requirements that are dictated by the disparate timing requirements for different parts of the system. For example the GPS receivers require 500 millisecond timing intervals, the ephemeris algorithm only operates on a 15-minute cycle due to the received ephemeris messages from the SVs, the corrections algorithm completes in 30 to 40 milliseconds, and the VDB transmission follows strict TDMA communication protocols on a 500 millisecond cycle. The parallelized and multithreaded software architecture developed in this work allows the various processes to run independently and meet each of their timing requirements. This was a major step in producing software that is capable of implementing the CL-GBAS algorithms in real-time for extended operation.

The JSON Data Logger class was created and employed to allow for the flexible collection of data at every step in the GBAS data pipeline into the highly common and portable JSON format. This was an essential feature that supports future research into the many, and varied, sub-algorithms and hardware changes one can conceive in continuing development of the GBAS.

The GPS Receiver class was implemented such that new subclasses could be created and seamlessly be inserted into the GBAS system. This allows developers to select an implement new receivers for the system, and to assess the effectiveness of the system if a diverse set of receivers is developed. A fundamental assumption in the GBAS literature is that a single receiver type is required for consistency. However, there is no body of research that has proven that, and the LVBS 3.0 developed here provides the substrate to research this avenue thoroughly, using the JSON Data Logger to intercept and analyze ranging and correction data at every step of the process.

The LabVIEW VDB communication process leverages the timing strength that LabVIEW offers to maintain a consistent broadcast period. The process also properly implements the preamble and wrappers to the Type 1 Message so that the communication protocol required by the VDB's internal process is maintained correctly for all messages traffic. This particular module of software can easily be extended to implement the Type 2 and Type 4 Messages as the system matures to that point.

5.2 Future Works

Future work on this project can extend the LVBS 3.0 system to fully implement the a GBAS that can transmit a signal-in-space to airborne users. The substrate for LAAS Type 2 and Type 4 Messages were developed in this work and are ideally suited for development in other research endeavors. The current structure provides a method through which future development of Type 2 and Type 4 Messages can be easily adjusted through a user interface (the Type 4 Message contains the approach information that is created for each specific approach provided to users). Future work is needed to incorporate the Far Field Monitor into the LVBS 3.0 system to fully implement the CL-GBAS concept and to monitor the integrity of the VDB transmissions. Studies to determine the MEDE and ideal probabilistic error terms for the system should be performed. The implementation of a system architecture that uses remote reference stations that communicate to the LGF over wireless communication would allow for more ideal placements of the RS antennas. The GPS Receiver class should be expanded to include more receivers and more receivers should be studied to evaluate their effectiveness in the GBAS application. Further system architecture changes could include running the primary LVBS 3.0 LabVIEW processes on a linux server as a service and using a windows or linux program that functions as a thin client user interface. Redundancy could be built into the LVBS 3.0 system by using more than four RS and automating RS fail-over in software. Test flights should be carried out to fully test the continuity, integrity and accuracy of LVBS 3.0 and to meet the test flight requirements of MASPS. The system could become more versatile by passively constructing RINEX files from the raw measurements that each RS captures allowing for easy surveying of the RS locations without the need of additional equipment. Additional algorithms can leverage the feedback provided by the CL-GBAS algorithm architecture such as an iterative approach to remove an SV from a solution or having the LVBS 3.0 system collect data from more than four RS stations but transmitting corrections from the combination of four RS that yield the most accurate Type 1 corrections.

Bibliography

[1] RTCA Inc., "GNSS-Based Precision Approach Local Area Augmentation System (LAAS) Signal-in-Space Interface Control Document (ICD)," July 2017, RTCA DO-246E.

[2] C. Davis, A. Archinal, J. Dyer, H. Wen, and J. Fagan, "Initial design and performance results of the university of oklahoma laas far-field monitor," in *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2007)*, 2007, pp. 379–385.

[3] Chad Davis, "Conceptualization and Implementation of a New and Novel Ground Based Augmentation System Utilizing Feedback Control," Dissertation, University of Oklahoma, Norman, Oklahoma, 2007.

[4] FAA.gov, "Ground-Based Navigation - Instrument Landing System (ILS)," Accessed: November 2, 2022. [Online]. Available: {https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gbng/ils}

[5] ——, "Ground-Based Navigation - Very High Frequency Omni-Directional Range (VOR)," Accessed: November 2, 2022. [Online]. Available: {https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gbng/vor}

[6] ——, "Satellite Navigation - GPS - How It Works," Accessed: November 2, 2022. [Online]. Available: {https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/gps/howitworks}

[7] ——, "Satellite Navigation — Ground Based Augmentation System (GBAS)," Accessed: November 2, 2022. [Online]. Available: {https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/laas}

[8] Alfred Leick, *GPS Satellite Surveying, Third Edition*. Hoboken, New Jersey: John Wiley & Sons, Inc, 2004.

[9] Pratap Misra and Per Enge, *Global Positioning System Signals, Measurements, and Performance, Second Edition*. Lincoln, Massachusetts: Ganga-Jamuna Press, 2006.

[10] Jonathan M. Harkness, “Mechanical Development of a Reference Receiver System for Use on the O.U. Local Area Augmentation System,” Thesis, University of Oklahoma, Norman, Oklahoma, 2009.

[11] John Dyer, “A New Siting Model For Reference Station Placement in a Ground Based Augmentation System,” Dissertation, University of Oklahoma, Norman, Oklahoma, 2008.

[12] Bradford W. Parkinson and James J. Skilker Jr., *Global Positioning System: Theory and Applications I*. Washington, DC: American Institute of Aeronautics and Astronaudics, Inc., 1996.

[13] Rick Pendergraft, “A New Adaptive Integrity Monitor For The Local Area Augmentation System Utilizing Closed Loop Feed Back,” Dissertation, University of Oklahoma, Norman, Oklahoma, 2013.

[14] Andrew A. Archinal, “Far-Field Monitoring of a Ground Based Augmentation System,” Thesis, University of Oklahoma, Norman, Oklahoma, 2006.

[15] Federal Aviation Administration, “Specification: Wide Area Augmentation System (WAAS),” September 1999, FAA-E-2892B.

[16] ——, “FAA WAAS System Overview,” May 2006, Image. [Online]. Available: {https://en.wikipedia.org/wiki/Wide_Area_Augmentation_System#/media/File:FAA_WAAS_System_Overview.jpg}

[17] Chad Sherrell, “Development of a Robust Ground Based Satellite Augmentation Software Architecture,” Thesis, University of Oklahoma, Norman, Oklahoma, 2006.

[18] Federal Aviation Administration, “Non-Fed Specification: Category I Local Area Augmentation System Ground Facility,” October 2005, FAA-E-AJW44-2937A.

[19] ——, “LAAS Architecture from FAA website,” May 2010, Image. [Online]. Available: {https://https://en.wikipedia.org/wiki/Local-area_augmentation_system#/media/File:LAAS_Architecture.svg}

[20] Ashtech Precision Products, *GG12 OEM Board Reference Manual, Revision A*, Santa Clara, California, July 2000.

[21] Nicholas Joseph Cannon, “Electrical Development of a Reference Receiver System For Use on the OU Local Area Augmentation System,” Thesis, University of Oklahoma, Norman, Oklahoma, 2009.

[22] Novatel, *User Guide GPS-702-GG, GPS-701-GG and GPS-702-GG-N*, November 2015.

[23] Jaxon Taylor, “Using Inexpensive Software-Defined Radios as GPS Receivers in a Ground-Based Augmentation System,” Thesis, University of Oklahoma, Norman, Oklahoma, 2022.

[24] GPS Networking, *ALDCBS1X2 GPS Aplified 1X2 Splitter Technical Product Data*.

[25] Telerad Aeronautical and Maritime Radiocommunication Systems, *Operational Manual VHF GBAS Transmitter EM9009 - EM9009A*, France, October 2015.

[26] ——, *Technical Manual VHF GBAS Receiver RE9009 - RE9009A*, France, October 2015.

[27] G. D. Priyadharshini and G. Suchitra, “Performance analysis of reed-solomon codes in digital communication system using labview,” March 2020.

[28] Telerad Aeronautical and Maritime Radiocommunication Systems, *VDB Series 90x9 GBAS Interface Control Document*, November 2003.

[29] RTCA Inc., “Minimum Aviation System Performance Standards for the Local Area Augmentation System (LAAS),” December 2004, RTCA DO-245A.

[30] Tony Anthony and Mary Kerns, “Interface Specification: Global Positioning System,” August 2022, IS-GPS-200N.

[31] J. Zhang and H. Gao, “Implementation of crc algorithm in uhf rfid test system based on labview,” in *2nd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2017)*. Atlantis Press, 2016, pp. 902–906.

Appendix A - List Of Symbols

(X_m, Y_m, Z_m)	ECEF Position of a Receiver Antenna
(X_n, Y_n, Z_n)	ECEF Position of an SV
$B_{PR}(n, m)$	Error Contribution from RS m to SV n
CE	Receiver Clock Error
C_{ic}	Amplitude of the Cosine of the Angle of Inclination Correction
C_{is}	Amplitude of the Sine of the Angle of Inclination Correction
C_{rc}	Amplitude of the Cosine of the Orbital Radius Correction
C_{rs}	Amplitude of the Sine of the Orbital Radius Correction
C_{uc}	Amplitude of the Cosine of the Argument of Latitude Correction
C_{us}	Amplitude of the Sine of the Argument of Latitude Correction

E_K	Eccentric Anomaly
HE	Horizontal Error
HOW	Hand-over-Word
HPL	Horizontal Protection Level
IDOT	Rate of Inclination Angle
IOD	Issue of Data
IODC	IOD Clock
IODE	IOD Ephemeris
K_{B_PR}	B-Value Threshold Term
LAL	Lateral Alert Limit
Lat	Latitude
Lon	Longitude
M	Mean Eccentric Anomaly

$MEDE$	Minimum Ephemeris Detection Error
M_0	Mean Eccentric Anomaly at Reference Time
M_n	Number of RS in S_m
M_{Kepler}	Mean Eccentric Anomaly from Kepler′s Equation
$M_{ephemeris}$	Mean Eccentric Anomaly from Ephemeris
M_{error}	Mean Eccentric Anomaly Error
Modified Z-Count	GPS Time in a 20 Minute Reference Frame
N	Number of Samples in Smoothed Pseudorange Filter
N_C	Number of Elements in S_C
$P - Value$	Ephemeris Decorrelation Parameter
PRC	Pseudorange Correction
PR_r	Raw Pseudorange
PR_s	Smoothed Pseudorange

PR_{SCA}	Carrier Smoothed and Receiver Clock Adjusted Pseudorange Correction
PR_{SC}	Smoothed Pseudorange Correction
PR_{corr}	Broadcast Pseudorange Correction
R	Range
RRC	Range Rate Correction
R_{raw}	Raw Range Measurement from GPS Receiver
$ReceiveTime$	Receive Time of Data
S	Filter Time Constant in Smoothed Pseudorange Filter
SAD	Source Availability Duration
S_C	Set of SVs Tracked by All RS
S_m	Set of RS Tracking SV n
S_n	Set of RS with Valid Measurements for SV n
T	Filter Sample Interval in Smoothed Pseudorange Filter

T_{GD}	Group Delay
VAL	Vertical Alert Limit
VE	Vertical Error
VPL	Vertical Protection Level
ΔX	Difference in ECEF Coordinates on the X-Axis
ΔY	Difference in ECEF Coordinates on the Y-Axis
ΔZ	Difference in ECEF Coordinates on the Z-Axis
Δn	Mean Anomaly Correction
Δt_r	Relativistic Clock Correction Term
Δt_{SV}	SV PRN Code Phase Time Offset
Ω_0	Longitude of the Ascending Node at Weekly Epoch
Ω_k	Corrected Longitude of the Ascending Node
Φ_k	Argument of Latitude

$\Theta(n)$	Elevation of SV n
δ_{i_k}	Angle of Inclination Correction
δ_{r_k}	Orbital Radius Correction
δ_{u_k}	Argument of Latitude Correction
$\hat{\Omega}$	Rate of Right Ascension
$\hat{\Omega}_e$	Earth's Rotation Rate
μ	Earth's Gravitational Constant for GPS
ω_n	Argument of Perigree
ϕ	Accumulated Phase Measurement
σ_{PR_GND}	LAAS Ground Error Bounding Term
$\sqrt{A}$	Square Root of Semi-Major Axis
a_{f0}	Ephemeris Clock Correction Coefficient 0
a_{f1}	Ephemeris Clock Correction Coefficient 1

a_{f2}	Ephemeris Clock Correction Coefficient 2
e	Eccentricity of Orbit
f_{L1}	L1 Frequency
i_0	Inclination Angle at Reference Time
i_k	Corrected Angle of Inclination
k	Index
m	Index of RS
n	PRN of SV
r_k	Corrected Orbital Radius
t_{SV_GPS}	Clock Correction
t_{oe}	Reference Time for Orbit
$t_{transmit}$	GPS System Time at Time of Transmission
u_k	Corrected Argument of Latitude

v_k	True Anomaly
x_{k_1}	Position of SV on the X-Axis of the Orbital Plane
x_{k_2}	Uncorrected ECEF Coordinate of SV on the X-Axis
$x_{k_{final}}$	ECEF Coordinate of SV on the X-Axis
y_{k_1}	Position of SV on the Y-Axis of the Orbital Plane
y_{k_2}	Uncorrected ECEF Coordinate of SV on the Y-Axis
$y_{k_{final}}$	ECEF Coordinate of SV on the Y-Axis
$z_{k_{final}}$	ECEF Coordinate of SV on the Z-Axis

Appendix B - List Of Acronyms and Abbreviations

1PPS	One Pulse Per Second
ASCII	American Standard Code for Information Interchange
Acronym	Definition
C/A Code	Coarse Acquisition Code
CL-GBAS	Closed Loop GBAS
CRC	Circular Redundancy Check
D8PSK	Differentially-Encoded 8 Phase Shift Keying
DAQ	Data Acquisition
DGPS	Differential GPS
DIN	Dual Inline Connector

ECEF	Earth Centered, Earth Fixed Coordinate System
FAS	Final Approach Segment
FEC	Forward Error Correction
FFM	Far Field Monitor
GBAS	Ground Based Augmentation System
GPS	Global Positioning System
GSL	GBAS Service Level
GUI	Graphical User Interface
HE	Horizontal Error
HPL	Horizontal Protection Level
ILS	Instrument Landing System
JSON	JavaScript Object Notation
L1	Link 1 Frequency

L2	Link 2 Frequency
LAAS	Local Area Augmentation System
LAL	Lateral Alert Limit
LGF	LAAS Ground Facility
LLA	Latitude, Longitude, Altitude Coordinate System
LM	Local Monitor
LVBS 3.0	LabVIEW Base Station 3.0
MASPS	Minimum Aviation System Performance Standards for LAAS
MCS	Master Control Station
MEDE	Minimum Ephemeris Detection Error
MIS	GG12 Miscellaneous Data Message
NSE	Navigational System Error Analysis
P(Y) Code	Precision Acquisition Code

PCB	Printed Circuit Board
PPS	Precise Positioning Service
PRC	Pseudorange Correction
PRN	Pseudo-random Number
RRC	Range Rate Correction
RS	Reference Station
SA	Selective Availability
SBAS	Space Based Augmentation System
SPS	Standard Positioning Service
SSID	Station Slot Identifier
SV	Space Vehicle
TDMA	Time Devision Multiple Access
TOA	Time of Arrival

VAL	Vertical Alert Limit
VDB	VHF Data Broadcast
VE	Vertical Error
VI	Virtual Instrument
VIM	Malleable VI
VOR	VHF Omni-directional Range
VPL	Vertical Protection Level
WAAS	Wide Area Augmentation System

Appendix C - Satellite Positioning and Validation Equations

C.1 Almanac Validation

New almanac data are considered valid under the following conditions:

a. Receiver communication completed with successful checksum

b. All health flags are set to 0 (healthy) for the SV

c. The PRN is in the expected range: PRN $\leq$ 32

C.2 Ephemeris Validation

New ephemeris data are validated against the existing ephemeris data if existing ephemeris data are available [18]. Ephemeris data are validated against the existing ephemeris by verifying that the difference in satellite position, calculated with the existing and new ephemeris is less than 250 meters [18]. New ephemeris data are validated against almanac data if ephemeris data for the SV are not available [3]. Ephemeris data must be valid continuously for two minutes before they are used to generate the LAAS solution. Ephemeris data must be validated and used within three minutes of being received.

C.2.1 Ephemeris Content Validation

The FAA-2937 requires that LAAS systems do not validate ephemeris data if any of the following conditions are met:

a. “Three or more parity errors have been detected from multiple receivers in the previous 6 seconds” [18].

b. ”Broadcast Issue of Data (IOD) Ephemeris (IODE) does not match eight least-significant bits of broadcast IOD Clock (IODC)” [18].

c. ”Bit 18 of the Hand-over-Word (HOW) is set to 1” [18].

d. ″All data bits are zeros in sub-frames 1, 2, or 3″ [18].

e. ″Default navigation data are being transmitted in sub-frames 1, 2, or 3″ [18].

f. ″The preamble does not equal 8B (hexadecimal)″ [18].

g. ″At any time in the next 12 hours, any point on the orbit defined by the broadcast ephemeris is more than 7000 m from the orbit defined by the broadcast alamanac″ [18]

h. ″Ephemeris CRC changes and IODE does not″ [18]

i. ″The PRN is 33, 34, 35, 36, or 37″ [18]

j. ″The health bits in sub-frame 1, word 3, indicate that the satellite is unhealthy″ [18]

Conditions a, c, d, e, and f are all directly available from the GG12 GPS receiver in the miscellaneous data message (MIS) [20]. Condition j is available directly from the ephemeris data from the GG12 GPS receiver [20]. Condition b is performed by extracting the least significant 8 bits from the IODC and then comparing it to the IODE. Condition g is verified by calculating the position of the SV based on almanac data and based on ephemeris data for a given SV at several times ranging from the current time to the time twelve hours in the future [3]. The distances between the almanac and ephemeris positions are calculated and considered valid if the distance is less than 7000 meters from each other [18]. Section C.3.1.1 describes how the positions of SVs are calculated. Condition h requires the CRC for the ephemeris data to be calculated. Section D.1.6 describes the process by which the CRC is generated. Conditions i and j can be validated directly from the ephemeris message from the GG12 receiver [20].

C.2.2 Ephemeris Timing Validation

After a new set of ephemeris data pass validation according to section C.2.1, the SV position is generated from the new ephemeris. The position of the SV is generated from the current ephemeris and then validated by checking that the distance between the current and new positions are less than 250 meters apart if ephemeris data are available. The elevation of the SV is calculated and the SV is considered invalid if the elevation is less than 5° over the horizon. The elevation calculation is described in section C.3.2. The validated ephemeris data are clustered with metadata that describe whether the new ephemeris data were able to be validated against the current ephemeris and the time that the new ephemeris data were received and validated. The validated ephemeris data with metadata cluster are inserted into a map with the SV PRN as the key.

Regardless of whether new ephemeris data were received during the current iteration of the algorithm while loop, the active ephemeris and the new ephemeris data with metadata cluster maps are processed. The both maps are validated every iteration by checking that the elevation of each SV is currently greater than 5° over the horizon. Elements from the new ephemeris data with metadata cluster are removed and inserted into the active ephemeris map when they have maintained validity for two minutes. The active ephemeris data are used to generate a map of other SV related data including SV elevation, ephemeris decorrelation parameter, and source availability duration for use in constructing the Type 1 messages. Details describing the ephemeris decorrelation parameter are in section D.1.5 and details describing the source availability duration are in section D.1.7.

C.3 Satellite Position and Elevation Algorithm

The satellite positioning algorithm subVI calculates the position of an SV, the eccentric anomaly of an SV (E_K), the range to the SV from a location, and the elevation of the SV from a location. The E_K term is calculated using an iterative method to solve Kepler's equation. The SV position, range, and E_K are all solved together in an iterative method that corrects for inaccuracy in the data transmit time initial condition [3]. The elevation calculation is performed once range and position are known. Equations are presented with angles in their native formats but angles are converted to radians in the software. Semicircles are a unit that is commonly used in GPS and it refers to the angle halfway around a circle, or π radians.

C.3.1 Kepler's Equation and Eccentric Anomaly ($\mathbf{E_K}$)

Kepler's equation describes the relationship between the mean eccentric anomaly (M) and E_K as shown in equation C.1 [8].

$$
\begin{aligned}
M_{Kepler}(k) &= E_K(k) - e \cdot \sin\left(E_K(k)\right) \qquad \text{(C.1)}\\
&\text{where}\\
M_{Kepler}(k) &= \text{Mean eccentric anomaly (semicircles)}\\
E_K(k) &= \text{Eccentric anomaly (semicircles)}\\
e &= \text{Eccentricity of orbit (semicircles)}\\
k &= \text{Index value}
\end{aligned}
$$

The $M(k)$ term can be calculated using the transmit time of the GPS data and parameters found in the ephemeris message. Equation C.2 describes how $M(k)$ is calculated.

$$M_{ephemeris} = M_0 + \left(\sqrt{\frac{\mu}{\left(\sqrt{A}\right)^6}} + \Delta n \right) \cdot (t_{transmit} - t_{oe}) \tag{C.2}$$

where

$M_{ephemeris}$ = Mean eccentric anomaly (semicircles)

M_0 = Mean eccentric anomaly at reference time (semicircles)

μ = Earth's gravitational constant for GPS user $\left(\frac{\text{meters}^3}{\text{second}^2}\right)$

$\sqrt{A}$ = Square root of semi-major axis $\left(\text{meters}^{(1/2)}\right)$

Δn = Mean anomaly correction $\left(\frac{\text{semicircles}}{\text{second}}\right)$

$t_{transmit}$ = GPS system time at time of transmission (seconds)

t_{oe} = Reference time for orbit (seconds)

k = Index value

The term $M_{error}(k)$ describes the error between the value of $M_{ephemeris}$ that is calculated from ephemeris data and the value of $M_{Kepler}(k)$ that is calculated from Kepler's equation. Equation C.3 describes how $M_{error}(k)$ is calculated.

$$M_{error}(k) = M_{ephemeris} - M_{Kepler}(k) \tag{C.3}$$

where

$M_{error}(k)$ = Mean eccentric anomaly error (semicircles)

$M_{ephemeris}$ = Mean eccentric anomaly from ephemeris data (semicircles)

$M_{Kepler}(k)$ = Mean eccentric anomaly from Kepler's equation (semicircles)

k = Index value

The algorithm used to determine $E_K(k)$ begins by calculating $M_{ephemeris}$ for an SV. The term $E_K(k)$ is initialized to $M_{ephemeris}$. Kepler's equation is used to solve for $E_K(k)$ as described in equation C.4 [3]. The term $M_{error}(k)$ is calculated using the new $E_K(k)$ term to solve for $M_{Kepler}(k)$ and the $M_{ephemeris}$ calculated during initialization [3]. The algorithm ends successfully when the term $M_{error}(k)$ is less than 10^{-14} or ends unsuccessfully if more than 20 iterations have passed.

$$E_K(k) = M_{ephemeris} + e \cdot \sin\left(E(k-1)\right) \tag{C.4}$$

where

$E_K(k)$ = Eccentric anomaly from current iteration (semicircles)

$E(k-1)$ = Eccentric anomaly from previouos iteration (semicircles)

$M_{ephemeris}$ = Mean eccentric anomaly (semicircles)

e = Eccentricity of orbit (semicircles)

k = Index value

C.3.1.1 Broadcast Navigation User Equation of Position

The position of an SV is calculated using an iterative algorithm that varies the GPS transmit time ($t_{transmit}(k)$) based on difference in the range from the SV to the receiver antenna in the current and previous iterations (R_{error}) [3]. The algorithm is initialized with by estimating the GPS transmit time and range based on the latest receive time and raw range as described in equations C.5 and C.6 respectively.

$$t_{transmit}(k-1) = \frac{ReceiveTime}{1000} - R_{raw} \tag{C.5}$$

$$R(k-1) = R_{raw} \tag{C.6}$$

where

$t_{transmit}(k-1)$ = GPS transmit time estimate of previous iteration (seconds)

$R(k-1)$ = Range estimate of previous iteration (seconds)

$ReceiveTime$ = Receive time of data (milliseconds)

R_{raw} = Range measurement from GPS receiver (seconds)

k = Index value

During each iteration of the algorithm the position of the SV is calculated from the ephemeris data. The $E_K(k)$ term is calculated using the method described in C.3.1. The true anomaly $v_k(k)$ is calculated as described in equation C.7 [30].

$$v_k(k) = 2\tan^{-1}\left(\sqrt{\frac{1+e}{1-e}} \cdot \tan\frac{E_K(k)}{2}\right) \tag{C.7}$$

where

$v_k(k)$ = True anomaly (semicircles)

e = Eccentricity of orbit (semicircles)

$E_K(k)$ = Eccentric anomaly from current iteration (semicircles)

k = Index value

The argument of latitude ($\Phi_k(k)$) and the second harmonic perturbations of latitude correction ($\delta u_k(k)$), orbital radius correction ($\delta r_k(k)$) and angle of inclination correction ($\delta i_k(k)$) are calculated according to the the equations C.8 - C.11. The argument of perigee (ω_n) and amplitudes of sines and cosines of the second harmonic perturbations are available directly from the ephemeris data [30].

$$\Phi_k(k) = v_k(k) + \omega_n \tag{C.8}$$
$$\delta u_k(k) = C_{us}\sin(2\Phi_k(k)) + C_{uc}\cos(2\Phi_k(k)) \tag{C.9}$$
$$\delta r_k(k) = C_{rs}\sin(2\Phi_k(k)) + C_{rc}\cos(2\Phi_k(k)) \tag{C.10}$$
$$\delta i_k(k) = C_{is}\sin(2\Phi_k(k)) + C_{ic}\cos(2\Phi_k(k)) \tag{C.11}$$

where

$\Phi_k(k)$ = Argument of latitude (semicircles)

$v_k(k)$ = True anomaly (semicircles)

ω_n = Argument of perigee (semicircles)

$\delta u_k(k)$ = Argument of latitude correction (radians)

$\delta r_k(k)$ = Orbital radius correction (meters)

$\delta i_k(k)$ = Angle of inclination correction (radians)

C_{us} = Amplitude of the sine of the argument of latitude correction (radians)

C_{uc} = Amplitude of the cosine of the argument of latitude correction (radians)

C_{rs} = Amplitude of the sine of the orbital radius correction (meters)

C_{rc} = Amplitude of the cosine of the orbital radius correction (meters)

C_{is} = Amplitude of the sine of the angle of inclination correction (radians)

C_{ic} = Amplitude of the cosine of the angle of inclination correction (radians)

k = Index value

The corrected argument of latitude ($u_k(k)$), corrected orbit radius ($r_k(k)$), and corrected inclination ($i_k(k)$) are calculated as described in equations C.12 - C.14

[30]. Terms $\sqrt{A}$, e, i_0, $IDOT$, and t_{oe} are all available directly from the ephemeris data message [20].

$$u_k(k) = \Phi_k(k) \cdot \pi + \delta u_k(k) \tag{C.12}$$

$$r_k(k) = \left(\sqrt{A}\right)^2 \cdot \, (1 - e \cdot \, \cos\left(E_K(k)\right)) + \delta r_k(k) \tag{C.13}$$

$$i_k(k) = (i_0 \cdot \pi) + (IDOT \cdot \pi)\left(t_{transmit}(k-1) - t_{oe}\right) + \delta i_k(k) \tag{C.14}$$

where

$u_k(k)$ = Corrected argument of latitude (radians)

$r_k(k)$ = Corrected orbital radius (meters)

$i_k(k)$ = Corrected angle of inclination (radians)

$\Phi_k(k)$ = Argument of latitude (semicircles)

$\delta u_k(k)$ = Argument of latitude correction (radians)

$\delta r_k(k)$ = Orbital radius correction (meters)

$\delta i_k(k)$ = Angle of inclination correction (radians)

$\sqrt{A}$ = Square root of semi-major axis $\left(\text{meters}^{(1/2)}\right)$

e = Eccentricity of orbit (semicircles)

$E_K(k)$ = Eccentric anomaly from current iteration (semicircles)

i_0 = Inclination angle at reference time (semicircles)

$IDOT$ = Rate of inclination angle $\left(\frac{\text{semicircles}}{\text{second}}\right)$

$t_{transmit}(k-1)$ = GPS transmit time estimate of previous iteration (seconds)

t_{oe} = Ephemeris data reference time of week (seconds)

k = Index value

The positions of the SV on the orbital plane $x_{k_1}(k)$ and $y_{k_1}(k)$ are calculated using equations C.15 and C.16 respectively [30].

$$x_{k_1}(k) = r_k(k) \cdot \cos(u_k(k)) \tag{C.15}$$

$$y_{k_1}(k) = r_k(k) \cdot \sin(u_k(k)) \tag{C.16}$$

where

$x_{k_1}(k)$ = Position of the SV on the x-axis of the orbital plane (meters)

$y_{k_1}(k)$ = Position of the SV on the y-axis of the orbital plane (meters)

$u_k(k)$ = Corrected argument of latitude (radians)

$r_k(k)$ = Corrected orbital radius (meters)

k = Index value

The corrected longitude of ascending node ($\Omega_k(k)$) is calculated as described in equation C.17 [30]. The longitude of the ascending note at weekly epoch (Ω_0) and the rate of right ascension ($\dot{\Omega}$) are available directly from the ephemeris message [20].

$$\Omega_k(k) = \Omega_0 \cdot \pi + \left(\dot{\Omega} \cdot \pi - \dot{\Omega}_e\right)(t_{transmit}(k-1) - t_{oe}) - \dot{\Omega}_e \cdot t_{oe} \tag{C.17}$$

where

$\Omega_k(k)$ = Corrected longitute of ascending node (radians)

Ω_0 = Longitude of the ascending node at weekly epoch (semicircles)

$\dot{\Omega}$ = Rate of right ascension $\left(\frac{\text{semicircles}}{\text{second}}\right)$

$\dot{\Omega}_e$ = Earth's rotation rate $\left(\frac{\text{radians}}{\text{second}}\right)$

$t_{transmit}(k-1)$ = GPS transmit time estimate of previous iteration (seconds)

t_{oe} = Ephemeris data reference time of week (seconds)

k = Index value

The earth centered, earth fixed (ECEF) coordinates of the SV ($x_{k_2}(k)$ and $y_{k_2}(k)$) are calculated as described in equations C.18 and C.19 [30].

$$x_{k_2}(k) = x_{k_1}(k) \cdot \cos(\Omega_k(k)) - y_{k_1}(k) \cdot \cos(i_k(k)) \cdot \sin(\Omega_k(k)) \tag{C.18}$$
$$y_{k_2}(k) = x_{k_1}(k) \cdot \sin(\Omega_k(k)) + y_{k_1}(k) \cdot \cos(i_k(k)) \cdot \cos(\Omega_k(k)) \tag{C.19}$$

where

$x_{k_2}(k)$ = ECEF coordinate of SV on x-axis (meters)

$y_{k_2}(k)$ = ECEF coordinate of SV on y-axis (meters)

$\Omega_k(k)$ = Corrected longitute of ascending node (radians)

$x_{k_1}(k)$ = Position of the SV on the x-axis of the orbital plane (meters)

$y_{k_1}(k)$ = Position of the SV on the y-axis of the orbital plane (meters)

$i_k(k)$ = Corrected angle of inclination (radians)

k = Index value

The ECEF coordinates $x_{k_2}(k)$ and $y_{k_2}(k)$ are corrected for Earth's rotation as described in equations C.20 and C.21 respectively [30]. The position of the SV is calculated as described in equation C.22. The final ECEF coordinates of the SV is represented by the vector $\left\langle x_{k_{final}}(k),\ y_{k_{final}}(k),\ z_{k_{final}}(k)\right\rangle$ [3].

$$x_{k_{final}}(k) = x_{k_2}(k) \cdot \cos\left(-\dot{\Omega}_e \cdot R(k-1)\right) - y_{k_2}(k) \cdot \sin\left(-\dot{\Omega}_e \cdot R(k-1)\right) \tag{C.20}$$
$$y_{k_{final}}(k) = x_{k_2}(k) \cdot \sin\left(-\dot{\Omega}_e \cdot R(k-1)\right) + y_{k_2}(k) \cdot \cos\left(-\dot{\Omega}_e \cdot R(k-1)\right) \tag{C.21}$$
$$z_{k_{final}}(k) = y_{k_1}(k) \cdot \sin(i_k(k)) \tag{C.22}$$

where

$x_{k_{final}}(k)$ = Final ECEF coordinate of SV on x-axis (meters)

$y_{k_{final}}(k)$ = Final ECEF coordinate of SV on y-axis (meters)

$z_{k_{final}}(k)$ = Final ECEF coordinate of SV on z-axis (meters)

$x_{k_2}(k)$ = Uncorrected ECEF coordinate of SV on x-axis (meters)

$y_{k_2}(k)$ = Uncorrected ECEF coordinate of SV on y-axis (meters)

$y_{k_1}(k)$ = Position of the SV on the y-axis of the orbital plane (meters)

$\dot{\Omega}_e$ = Earth's rotation rate $\left(\frac{\text{radians}}{\text{second}}\right)$

$R(k-1)$ = Range estimate of previous iteration (seconds)

$i_k(k)$ = Corrected angle of inclination (radians)

k = Index value

The range between the SV position and the surveyed position of a GPS receiver antenna $R(k)$ is calculated according to the method described in section D.1.10.1. The algorithm calculates R_{error} according to equation C.23 and stops if $R_{error} < 10^{-14}$, otherwise the algorithm iterates again with the new $R(k)$ and calculates a new $t_{transmit}$ according to equation C.24 [3].

$$R_{error} = R(k) - R(k-1) \tag{C.23}$$

$$t_{transmit}(k) = \frac{ReceiveTime}{1000} - R(k) \tag{C.24}$$

where

R_{error} = Range error between iterations (seconds)

$R(k)$ = Range estimate of current iteration (seconds)

$R(k-1)$ = Range estimate of previous iteration (seconds)

$t_{transmit}(k)$ = GPS transmit time estimate of current iteration (seconds)

$ReceiveTime$ = Receive time of data (milliseconds)

k = Index value

C.3.2 Elevation Calculation

The elevation of an SV is calculated from the ECEF and Latitude, Longitude, Altitude (LLA) positions of the SV and the GPS receiver antenna as described in equation C.25 [3]. The range between the SV and the GPS receiver (R) is calculated according to the method described in section D.1.10.1.

$$EL = \frac{\arcsin\left(\Delta X \cos(Lat)\cos(Lon) + \Delta Y \cos(Lat)\sin(Lon) + \Delta Z \sin(Lat)\right)}{R} \tag{C.25}$$

where

ΔX = The difference in ECEF coordinate x of the SV and receiver (meters)

ΔY = The difference in ECEF coordinate y of the SV and receiver (meters)

ΔZ = The difference in ECEF coordinate z of the SV and receiver (meters)

Lat = Surveyed latitude of the receiver (degrees)

Lon = Surveyed longitude of the receiver (degrees)

R = Range estimate from receiver to SV (meters)

Appendix D - GBAS and CL-GBAS Equations

D.1 Type 1 Message and Processing Algorithm

D.1.1 Modified Z-count

The Modified Z-count correlates with the GPS time that measurements were taken and resets every 20 minutes starting on the hour [18]. The Modified Z-count is calculated according to equation D.1. The GPS receive time is stored in LVBS 3.0 in milliseconds and Modified Z-count is represented in seconds.

$$\begin{aligned} \text{Modified Z-count} \equiv\ & (ReceiveTime \mod t_{20m})\,/1000 \qquad \text{(D.1)} \\ \text{where} & \\ ReceiveTime =\ & \text{Receive time of data (milliseconds)} \\ t_{20m} =\ & \text{Milliseconds in 20 minutes. (1200000 milliseconds)} \end{aligned}$$

D.1.2 Additional Message Flag

The additional measurement flag indicates if the Type 1 message will be sent in a single time slot in the TDMA frame or if the measurement block will be divided between multiple time slots within the same TDMA frame [1]. If multiple time slots are used the two Type 1 messages will contain at least one measurement block each and will share the same Modified Z-count. Table D.1 shows the possible values for the Additional Message Flag [1]. In LVBS 3.0 only a single message is used, therefore the Additional Message Flag is always 0 [18].

Table D.1 Additional Message Flag Values

Value	Description
0	All measurement blocks for a particular measurement type are contained in a single Message Type 1.
1	This is the first transmitted message of a linked pair of Type 1 Messages that together contain the set of all measurement blocks for a particular measurement type.
2	Reserved.

Table D.1 (*cont*)	
3	This is the second transmitted message of a linked pair of Type 1 Messages that together contain the set of all measurement blocks for a particular measurement type.

Table D.1. ***Additional Message Flag Values**: In LVBS 3.0 only single Type 1 messages are sent therefore 0 is always used [1].*

D.1.3 Number of Measurements

The Number of Measurements field describes how many SVs are present in the Type 1 measurement block [18]. The Number of Measurements field is calculated after the measurement block is completed by finding the size of the measurement block map.

D.1.4 Measurement Type

The measurement type indicates the type of ranging signal that provides the basis for the measurement block [18]. Table D.2 shows the potential values of Measurement Type. LVBS 3.0 only uses C/A code L1 therefore the value is always 0.

Table D.2 Measurement Type Flag Values	
Value	Description
0	C/A code L1
1	C/A code L2
2	P(Y) code L1
3	P(Y) code L2
4 - 7	Reserved

Table D.2. ***Measurement Type Flag Values**: In LVBS 3.0 only C/A code L1 is used therefore 0 is always used [1].*

D.1.5 Ephemeris Decorrelation Parameter (P-Value)

The Ephemeris Decorrelation Parameter (P-Value) describes "the impact of residual ephemeris errors due to spatial decorrelation for the ranging source, associated with the first ranging source measurement block" [18]. The P-Value for a given SV is calculated by dividing the minimum ephemeris detection error (MEDE) by the pseudorange between the SV and the LGF in meters. SVs with P-Values greater than $1.5 \times 10^{-4} \text{meters}/\text{meter}$ are excluded from use in the GBAS algorithms. The value of MEDE requires further study to determine the value for the OU LAAS system and depends on the method used to validate the ephemeris of a given SV [3]. Table D.3 shows the values of the MEDE used for different validation methods. Equation D.2 shows how to calculate a P-Value.

Table D.3 MEDE Values	
Validation Method	MEDE
Almanac Only	$3000m$
Almanac and Ephemeris	$1900m$

Table D.3. ***MEDE Values***: *The MEDE is determined based on what validation method was used to validate the ephemeris of an SV.*

$$P - Value_n \left(\frac{m}{m}\right) = \frac{MEDE(m)}{PR_n(m)} \tag{D.2}$$

D.1.6 Ephemeris Cyclic Redundancy Check (CRC)

A circular redundancy check (CRC) is an error correction method where a CRC checksum is generated and appended to data before transmission and can be used by the recipient to detect errors accumulated during transmission [31]. The CRC is performed on words three through ten of subframes one through three of the data transmission of an SV [18]. The Ephemeris CRC is generated by first constructing the raw ephemeris data structure, applying a bit mask to the raw ephemeris data, and then performing a 16-bit cyclic redundancy check algorithm. The ephemeris data from the GG12 receiver contains all of the information necessary to reconstruct the raw ephemeris data but does not directly output the raw ephemeris data. Table D.4 shows the ephemeris mask used to isolate the ephemeris data from the raw GPS message [1]. Equation D.3 defines the generator polynomial used for the ephemeris CRC [1].

$$G(x) = x^{16} + x^{12} + x^5 + 1 \tag{D.3}$$

Table D.4 Ephemeris CRC Mask				
Word	Byte	Ephemeris Parameter	Bits	Mask
Subframe 1:				
3	1-2	-	-	00000000
3	3	$IODC$	9-8	00000011
4	1-3	-	-	00000000
5	1-3	-	-	00000000
6	1-3	-	-	00000000
7	1-2	-	-	00000000
7	3	T_{GD}	7-0	11111111
8	1	$IODC$	7-0	11111111
8	2	t_{oc}	15-8	11111111
8	3	t_{oc}	7-0	11111111
9	1	a_{f2}	7-0	11111111

Table D.4 (*cont*)

Word	Byte	Ephemeris Parameter	Bits	Mask
9	2	a_{f1}	15-8	11111111
9	3	a_{f1}	7-0	11111111
10	1	a_{f0}	21-14	11111111
10	2	a_{f0}	13-6	11111111
10	3	a_{f0}	5-0	11111100
Subframe 2:				
3	1	$IODE$	7-0	11111111
3	2	C_{rs}	15-8	11111111
3	3	C_{rs}	7-0	11111111
4	1	Δn	15-8	11111111
4	2	Δn	7-0	11111111
4	3	M_0	31-24	11111111
5	1	M_0	23-16	11111111
5	2	M_0	15-8	11111111
5	3	M_0	7-0	11111111
6	1	C_{UC}	15-8	11111111
6	2	C_{UC}	7-0	11111111
6	3	e	31-24	11111111
7	1	e	23-16	11111111
7	2	e	15-8	11111111
7	3	e	7-0	11111111
8	1	C_{US}	15-8	11111111
8	2	C_{US}	7-0	11111111
8	3	$\sqrt{A}$	31-24	11111111
9	1	$\sqrt{A}$	23-16	11111111
9	2	$\sqrt{A}$	15-8	11111111
9	3	$\sqrt{A}$	7-0	11111111
10	1	t_{oe}	15-8	11111111
10	2	t_{oe}	7-0	11111111
10	3	-	-	00000000
Subframe 3:				
3	1	C_{ic}	15-8	11111111
3	2	C_{ic}	7-0	11111111
3	3	Ω_0	31-24	11111111
4	1	Ω_0	23-16	11111111
4	2	Ω_0	15-8	11111111
4	3	Ω_0	7-0	11111111
5	1	C_{is}	15-8	11111111
5	2	C_{is}	7-0	11111111

Table D.4 (*cont*)

Word	Byte	Ephemeris Parameter	Bits	Mask
5	3	i_0	31-24	11111111
6	1	i_0	23-16	11111111
6	2	i_0	15-8	11111111
6	3	i_0	7-0	11111111
7	1	C_{rc}	15-8	11111111
7	2	C_{rc}	7-0	11111111
7	3	ω	31-24	11111111
8	1	ω	23-16	11111111
8	2	ω	15-8	11111111
8	3	ω	7-0	11111111
9	1	$\dot{\Omega}$	23-16	11111111
9	2	$\dot{\Omega}$	15-8	11111111
9	3	$\dot{\Omega}$	7-0	11111111
10	1	*IODE*	7-0	11111111
10	2	*IDOT*	13-6	11111111
10	3	*IDOT*	5-0	11111100

Table D.4. ***Ephemeris CRC Mask***: *The ephemeris mask is applied via an AND operation. Appendix A defines the parameters listed in this table. The bits are in order of most significant (left) to least significant (right).*

D.1.7 Source Availability Duration (SAD)

The Source Availability Duration is the amount of time in seconds until an SV is below 5 degrees of elevation [18]. The SAD is required to be accurate within ±60 seconds [18]. The maximum transmittable value for the SAD is 2540 seconds [1]. The SAD from a ground facility that does not provide the SAD is coded as hexadecimal FF. The SAD is determined via the bisection method (an iterative root finding algorithm) if the elevation of an SV is currently above 5 degrees of elevation and it will be below 5 degrees of elevation within three hours. Figure D.1 shows the algorithm used to determine the SAD for an SV.

D.1.8 Ranging Source ID (PRN)

The pseudorandom number (PRN) is used as an ID for each SV [18]. LVBS 3.0 uses the PRN to indicate which data to associate with each SV. The PRN is present in all almanac, ephemeris, and measurement messages from the GG12 GPS receiver [20].

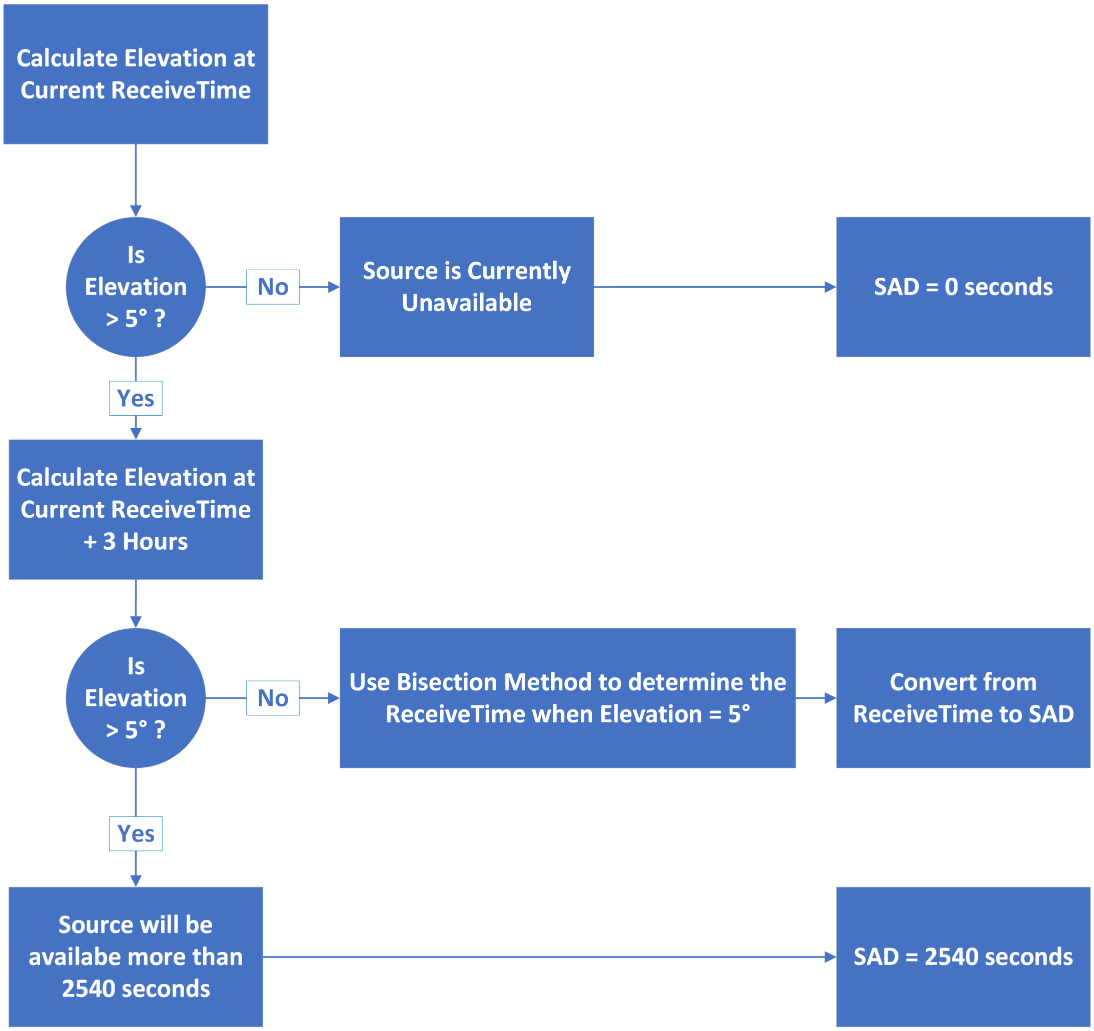

Calculate Elevation at Current ReceiveTime
Is Elevation > 5° ?
No
Source is Currently Unavailable
SAD = 0 seconds
Yes
Calculate Elevation at Current ReceiveTime + 3 Hours
Is Elevation > 5° ?
No
Use Bisection Method to determine the ReceiveTime when Elevation = 5°
Convert from ReceiveTime to SAD
Yes
Source will be availabe more than 2540 seconds
SAD = 2540 seconds

Figure D.1. ***Source Availability Duration Algorithm Diagram***: *The elevation is first evaluated at the current receive time. If the elevation is less than 5° then the SV is treated as unavailable and the SAD is set to 0 seconds. If the first elevation is greater than 5° then the elevation is evaluated at the current receive time + three hours. If the elevation is greater than 5° then the SAD is treated as available for the entire duration and the SAD is set to the maximum value 2540 seconds. If the elevation is less than 5° then a bisection algorithm is used to estimate when the elevation will be less than 5°.*

D.1.9 Issue of Data (IOD)

The IODE is the Issue of Data of Ephemeris [18]. The Issue of Data (IOD) field of the Type 1 message denotes the IODE and can be taken directly from the ephemeris messages from the GG12 GPS receiver [20].

D.1.10 Pseudorange Correction (PRC)

The Pseudorange Correction (PRC) is defined as the average of corrections to the pseudorange of the source signal in space based on the carrier smoothed code pseudorange measurements from multiple ground reference receivers [1]. The PRC is calculated following the steps listed in sections D.1.10.1 - D.1.10.5.

D.1.10.1 Predicted Range (R)

The Predicted Range is the distance between the surveyed position of each GPS antenna and the calculated position of each SV [18]. Equation D.4 defines the calculation used in LVBS 3.0 [3].

$$R(n,m) = \frac{\sqrt{(X_n - X_m)^2 + (Y_n - Y_m)^2 + (Z_n - Z_m)^2}}{C} \tag{D.4}$$

where

R = Predicted Range (seconds)

(X_n, Y_n, Z_n) = Calculated position of SV (ECEF)

(X_m, Y_m, Z_m) = Surveyed position of a receiver antenna (ECEF)

n = PRN of SV

m = Index of receiver antenna

C = Speed of light constant $\left(\frac{meters}{second}\right)$

D.1.10.2 Smoothed Pseudorange (PR_S)

The Smoothed Pseudorange is calculated for each SV that each receiver obtains measurements from. Phase measurements must be converted from cycles to seconds during calculations because the GG12 GPS receiver outputs phase measurements only in cycles [20]. Equation D.5 shows the conversion of phase from cycles to seconds.

$$\phi(n,m,k) = \frac{\Phi(n,m,k)}{f_{L1}} \tag{D.5}$$

where

$\phi(n,m,k)$ = Accumulated phase (seconds)

$\Phi(n,m,k)$ = Accumulated phase (cycles)

f_{L1} = Carrier frequency of L1 signal (Hz)

n = PRN of SV

m = Index of receiver antenna

k = Index of measurement

Each time new range measurements from an SV to a receiver are acquired the accumulated phase, raw pseudorange, and calculated smoothed pseudorange are recorded and a feedback node is used to retain the values for the next iteration. The variable $N(n,m)$ is used to record the number of valid measurements that have been received in a row with a maximum limit of 200 measurements. When conditions for invalidating a measurement are met, $N(n,m)$ is reset to 0. The measurements are considered invalid if the accumulated phase changes by more than 0.1% between samples or if the accumulated phase or raw pseudorange are not received within 0.5 seconds [29]. The smoothed pseudorange is calculated according to equation D.6 [18].

$$\begin{aligned} PR_S(n,m,k) = \left(\frac{PR_r(n,m,k)}{N(n,m)}\right) + \left(\frac{N(n,m)-1}{N(n,m)}\right)[PR_S(n,m,k-1) \\ + \phi(n,m,k) - \phi(n,m,k-1)] \end{aligned} \tag{D.6}$$

where

$PR_r(n,m,k)$ = Raw pseudorange (seconds)

$PR_S(n,m,k)$ = Smoothed pseudorange (seconds)

$N(n,m)$ = Number of samples

$\phi(n,m,k)$ = Accumulated phase measurement (seconds)

n = PRN of SV

m = Index of receiver antenna

k = Index of measurement

D.1.10.3 Clock Correction (t_{SV_GPS})

The Clock Correction t_{SV_GPS} is a term used to correct the deterministic SV clock error characteristics of bias, drift, and aging as well as the SV implementation characteristics of group delay bias and mean differential group delay [30]. The

SV PRN code phase offset (Δt_{SV}) is calculated from parameters obtained from the ephemeris data for each SV. The group delay term (T_{GD}) is obtained directly from the GG12 GPS receiver. Equation D.7 describes the relationship between t_{SV_GPS}, Δt_{SV} and T_{GD}.

$$t_{SV\_GPS}(n,m) = \Delta t_{SV}(n,m) - T_{GD}(n) \tag{D.7}$$

$$\begin{aligned} &\text{where} \\ t_{SV\_GPS}(n,m) &= \text{Clock correction (seconds)} \\ \Delta t_{SV}(n,m) &= \text{SV PRN code phase time offset (seconds)} \\ T_{GD}(n) &= \text{Group delay (seconds)} \\ n &= \text{PRN of SV} \\ m &= \text{Index of receiver antenna} \end{aligned}$$

The Δt_{SV} term is calculated using clock correction coefficients (a_{f_0}, a_{f_1} and a_{f_2}), the clock data reference time (t_{oc}), and the relativistic correction term (Δt_r). The relativistic correction term is calculated using the eccentricity (e) and the square root of the semi-major axis ($\sqrt{A}$) from the ephemeris as well as the eccentricity anomaly (E_k) that is calculated as described in section C.3.1. Equation D.8 describes how Δt_r is calculated. Equation D.9 describes how Δt_{SV} is calculated [30].

$$\Delta t_r(n,m) = F \cdot e(n) \cdot \left(\sqrt{A}(n)\right) \cdot \sin\left(E_k(n,m)\right) \tag{D.8}$$

$$\begin{aligned} &\text{where} \\ F &= \frac{-2\sqrt{\mu}}{C^2}, \text{ constant value } \left(\frac{\text{seconds}}{\sqrt{\text{meter}}}\right) \\ \mu &= \text{Earth's gravitational parameter } 3.986005 \times 10^{14} \left(\frac{\text{meters}^3}{\text{second}^2}\right) \\ C &= \text{Speed of light } \left(\frac{\text{meters}}{\text{second}}\right) \\ e(n) &= \text{Eccentricity} \\ \sqrt{A}(n) &= \text{Square root of semi-major axis (meters}^{1/2}) \\ E_k(n,m) &= \text{Eccentricity anomaly} \\ n &= \text{PRN of SV} \\ m &= \text{Index of receiver antenna} \end{aligned}$$

$$\begin{aligned}\Delta t_{SV}(n,m) = a_{f0}(n) + a_{f1}(n) \cdot (t(n,m) - t_{oc}(n)) \\ + a_{f2}(n) \cdot (t(n,m) - t_{oc}(n))^2 + \Delta t_r(n,m)\end{aligned} \tag{D.9}$$

$$\begin{aligned}
&\textit{where} \\
a_{f0}(n) &= \text{Clock correction coefficient 0 (seconds)} \\
a_{f1}(n) &= \text{Clock correction coefficient 1 } \frac{\text{seconds}}{\text{second}} \\
a_{f2}(n) &= \text{Clock correction coefficient 2 } \frac{\text{seconds}}{\text{second}^2} \\
t(n,m) &= ReceiveTime(n,m) - R(n,m), \text{ Estimate transmit time (seconds)} \\
ReceiveTime(n) &= GPSdatareceivetime(seconds) \\
R(n,m) &= \text{Range to SV from Antenna (seconds)} \\
t_{oc}(n) &= \text{Clock reference time (seconds)} \\
\Delta t_r(n,m) &= \text{Relativistic Correction Term (seconds)} \\
n &= \text{PRN of SV} \\
m &= \text{Index of receiver antenna}
\end{aligned}$$

D.1.10.4 Smoothed Pseudorange Correction (PR$_{SC}$)

The Smoothed Pseudorange Correction (PR$_{SC}$) is generated for each PR$_S$ that is calculated and passes validation as described in section D.1.10.2. Each PR$_{SC}$ that is generated is validated by checking the magnitude of the correction is less than 100 meters [3]. Equation D.10 describes how the PR$_{SC}$ is calculated [18].

$$PR_{SC} = R(n,m) - PR_s(n,m) - t_{SV\_GPS}(n,m) \tag{D.10}$$

$$\begin{aligned}
&\text{where} \\
R(n,m) &= \text{Predicted Range (seconds)} \\
PR_s(n,m) &= \text{Smoothed Pseudorange (seconds)} \\
t_{SV\_GPS}(n,m) &= \text{Clock Correction (seconds)} \\
n &= \text{PRN of SV} \\
m &= \text{Index of receiver antenna}
\end{aligned}$$

D.1.10.5 Broadcast Pseudorange Correction (PRC)

The broadcast PRC is calculated using only the SVs that are being tracked by all four of the GPS receivers unless one of the receivers is being excluded. A receiver will be excluded if it limits total the number of shared SVs to less than four. A receiver will be excluded if it lowers the number of shared SVs by two

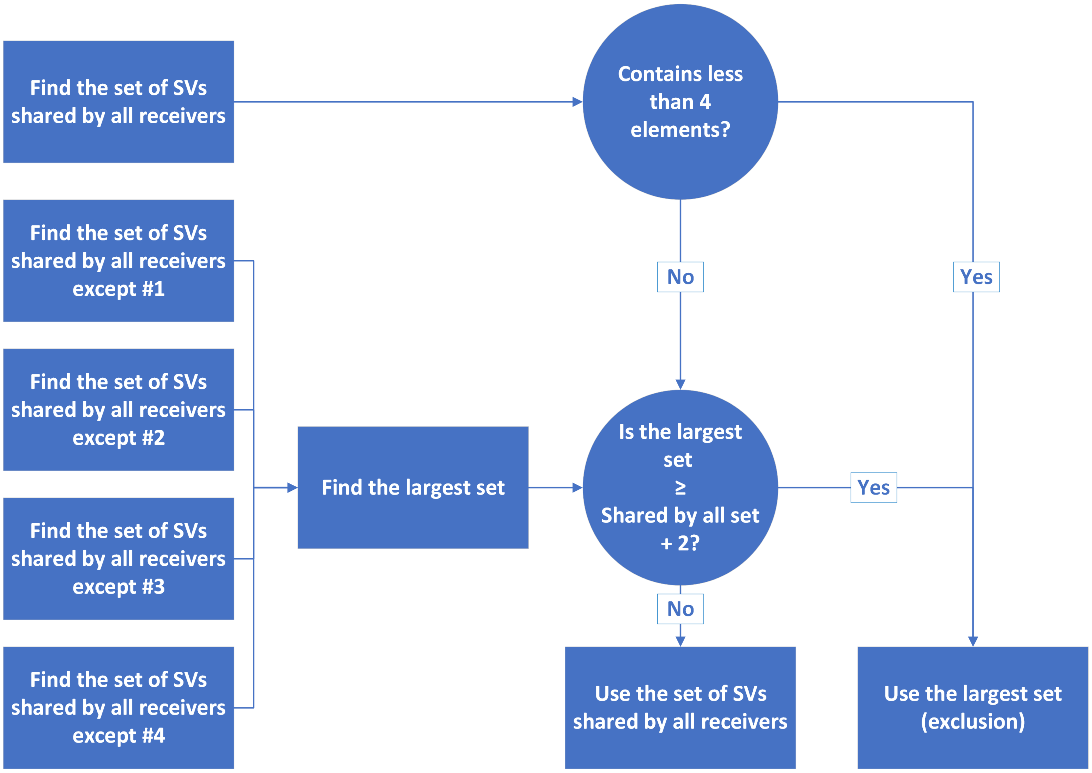

Find the set of SVs shared by all receivers
Contains less than 4 elements?
Find the set of SVs shared by all receivers except #1
Find the set of SVs shared by all receivers except #2
Find the set of SVs shared by all receivers except #3
Find the set of SVs shared by all receivers except #4
Find the largest set
No
Yes
Is the largest set ≥ Shared by all set + 2?
Yes
No
Use the set of SVs shared by all receivers
Use the largest set (exclusion)

Figure D.2. ***Reference Station Exclusion Algorithm Diagram:*** *A receiver may be excluded from the PRC calculation if the set of SVs shared by all receivers is less than 4 elements or if the largest set of SVs shared by any 3 receivers is 2 elements larger than the set of SVs shared by all receivers.*

when included in the solution [3]. The broadcast PRC is not calculated if the set of SVs that would be used contains less than four SVs [18]. Figure D.2 describes the algorithm used to exclude a receiver from the PRC calculation.

The carrier smoothed and receiver clock adjusted PRC (PR_{SCA}) is calculated for each valid PR_{SC} that is generated for each SV being tracked from each GPS receiver. The PR_{SCA} is calculated by subtracting the receiver clock error (CE) from each of the corresponding PR_{SC} [18]. The receiver clock error is calculated by averaging the PR_{SC} values generated from each receiver for each satellite that will be used [18]. A different CE is calculated for each receiver. Equation D.11 describes how the CE is calculated [3]. Equation D.12 shows how the CE is used to calculate the PR_{SCA} [18].

$$CE(m) = \frac{1}{N_C} \sum_{n \in S_C} PR_{SC}(n, m) \tag{D.11}$$

where

$CE(m)$ = Receiver clock error (seconds)

$PR_{SC}(n, m)$ = Smoothed Pseudorange Correction (seconds)

S_C = The set of all ranging sources that will be used

N_C = The number or elements in set S_C

n = PRN of SV

m = Index of receiver antenna

$$PR_{\text{SCA}} = PR_{\text{SC}} - CE(m) \tag{D.12}$$

where

$PR_{SCA}(n, m)$ = Carrier Smoothed & receiver clock adjusted PRC (seconds)

$PR_{SC}(n, m)$ = Smoothed Pseudorange Correction (seconds)

$CE(m)$ = Receiver clock error (seconds)

n = PRN of SV

m = Index of receiver antenna

The Broadcast Pseudorange Correction (PRC) is calculated for each SV that is being used. The PRC is the average of the PR_{SCA} values calculated from each GPS receiver for a given SV. Equation D.13 describes the calculation used to calculate the PRC [18].

$$PRC(n) = \frac{1}{M_n} \sum_{m \in S_m} PR_{SCA}(n, m) \tag{D.13}$$

where

$PRC(n)$ = Broadcast Pseudorange Correction (meters)

$PR_{SCA}(n, m)$ = Carrier Smoothed & receiver clock adjusted PRC (meters)

S_m = Set of receivers that are tracking the SV

M_n = Number or receivers in set S_m

n = PRN of SV

m = Index of receiver antenna

D.1.11 Range Rate Correction (RRC)

The Range Rate Correction (RRC) is the rate of change of the PRC. The RRC is calculated for each SV for which a PRC has been generated. The SV shall be excluded from the measurement block if the RRC exceeds $\pm 3.4 \left(\frac{\text{meters}}{\text{second}}\right)$ [18]. Equation D.14 describes how the RRC is calculated for each SV [18].

$$RRC(n,k) = \frac{PRC(n,k) - PRC(n,k-1)}{ReceiveTime(k) - ReceiveTime(k-1)} \tag{D.14}$$

where

$$\begin{aligned} RRC(n,k) &= \text{Range Rate Correction } \left(\frac{\text{meters}}{\text{second}}\right) \\ PRC(n,k) &= \text{Pseudorange Correction (meters)} \\ ReceiveTime(k) &= \text{GPS data receive time (seconds)} \\ n &= \text{PRN of SV} \\ k &= \text{Index of measurement} \end{aligned}$$

D.1.12 Sigma Pseudorange Ground ($\sigma_{\mathbf{PR\_GRND}}$)

The Sigma Pseudorange Ground ($\sigma_{\text{PR\_GRND}}$) term bounds the error contribution from the LAAS ground system. The upper limit of the $\sigma_{\text{PR\_GRND}}$ is a function of the elevation of each SV and the ground accuracy designator assigned to each GPS receiver. The ground accuracy designator assigned to a LAAS system is a combination of a character and a number. The character describes the accuracy of the GPS receivers used in the LAAS system. The number describes the number of GPS receivers used in the LAAS solution. Table D.5 defines the ground accuracy designator characters [29].

Table D.5 Ground Accuracy Designators

Ground Accuracy Designator	Definition
A	Represents an accuracy standard achievable using commonly available receivers and modest multipath mitigation techniques.
B	Represents an improved accuracy consistent with the use of higher accuracy modern receivers and better multipath mitigation techniques.
C	Represents an accuracy consistent with state of the art GPS reciever and multipath mitigation technologies.

Table D.5. ***Ground Accuracy Designators**: The ground accuracy designator character is a function of the quality of the GPS receivers used in the LAAS system. [29]*

The upper limit of $\sigma_{\text{PR\_GRND}}$ for GPS only is defined in equation D.15. Table D.5 defines the parameters used to calculate the upper limit of $\sigma_{\text{PR\_GRND}}$ for each ground accuracy designator.

$$\sigma_{PR\_GND}(\theta(n)) \leq \sqrt{\frac{(a_0 + a_1 \cdot e^{-\theta(n)/\theta_0})^2}{M_n} + (a_2)^2 + \left(\frac{a_3}{\sin(\theta(n))}\right)^2} \tag{D.15}$$

$\sigma_{PR\_GND}(\theta(n))$ = Bound of error contribution by LAAS ground system (meters)

$\theta(n)$ = Elevation of SV

n = PRN of SV

S_m = Set of receivers that are tracking the SV

M_n = Number or receivers in set S_m

Table D.6 Sigma Pseudorange Ground Upper Limit Parameters						
Ground Accuracy Designator	$\theta(n)$	θ_0	a_0	a_1	a_2	a_3
Units	Degrees	Degrees	Meters	Meters	Meters	Meters
A	> 5	14.3	0.5	1.65	0.08	0.03
B	> 5	15.5	0.16	1.07	0.08	0.03
C	> 35	15.5	0.15	0.84	0.04	0.01
C	≤ 35	-	0.24	0	0.04	0.01

Table D.6. ***Sigma Pseudorange Ground Upper Limit Parameters**: Parameters used to calculate the upper limit of $\sigma_{\text{PR\_GRND}}$ for different ground accuracy designators. The selection of the parameters is a function of the elevation of the SV. [29]*

The values used for $\sigma_{\text{PR\_GRND}}$ at different elevations are configurable in LVBS 3.0. The validation criteria of B-Values is dependent upon the $\sigma_{\text{PR\_GRND}}$. Lowering the $\sigma_{\text{PR\_GRND}}$ affects the LAAS availability because the B-Values become more likely to be invalidated but decrease the risk present in the system. This is reflected in the vertical and horizontal protection levels (described in section D.2.7), that are used to validate signal quality in CL-GBAS. More research is required to determine criteria for setting $\sigma_{\text{PR\_GRND}}$ in CL-GBAS. The validation criteria for the B-Values is defined in section D.1.13. During standard operation LVBS 3.0 is configured to use $\sigma_{\text{PR\_GRND}} = 0.2$ for all elevations with three or four GPS receivers. This classifies the ground accuracy designator for LVBS3.0 as A3

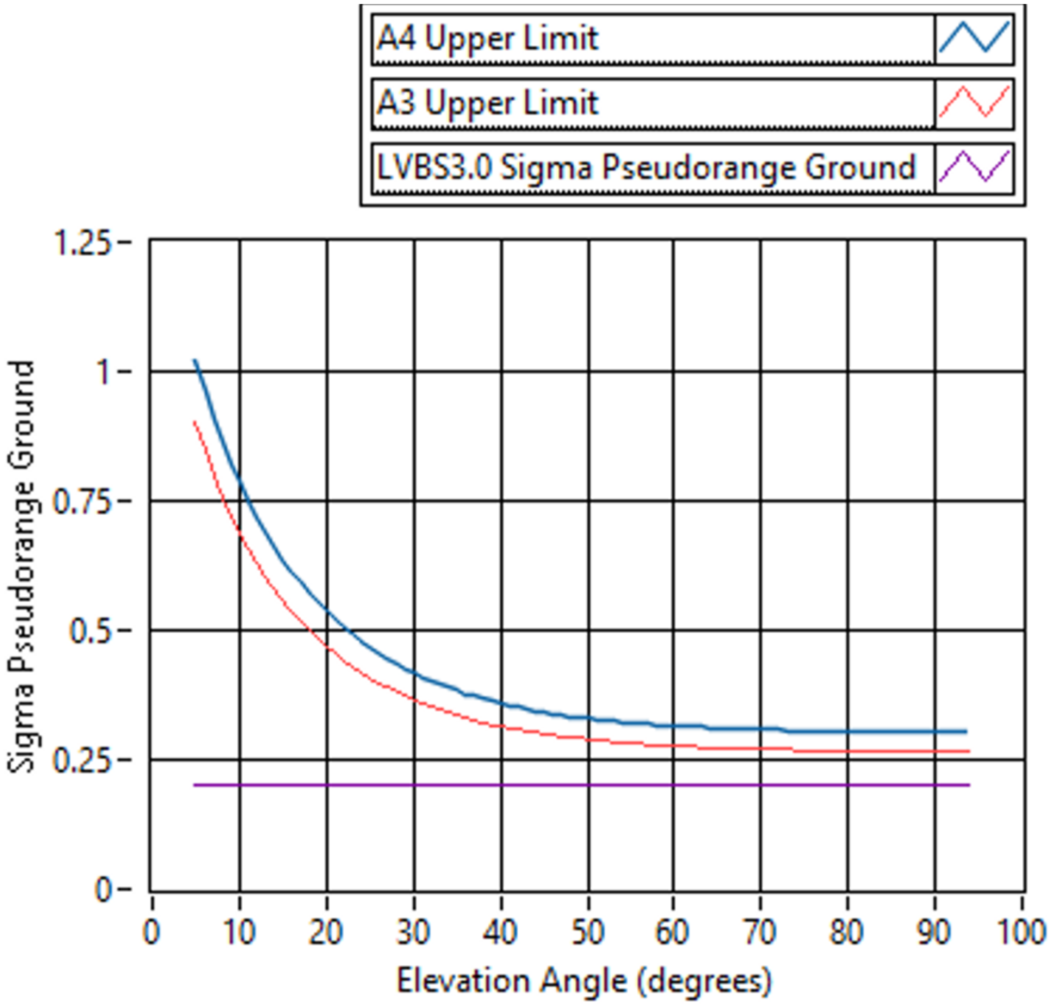


Figure D.3. ***Sigma Pseudorange Ground Upper Limits***: *The value of $\sigma_{\text{PR\_GRND}}$ is lower than the upper limits of the A4 and A3 ground accuracy designators for all elevations*

or A4. Using $\sigma_{\text{PR\_GRND}}$ below the maximum limit ground accuracy designators A3 and A4 allows the LVBS3.0 system to maintain a balance of availability and integrity. Figure D.3 shows plots of the maximum limits for A3 and A4 with the $\sigma_{\text{PR\_GRND}}$ used in LVBS3.0.

D.1.13 B-Values

Up to four B-Values are calculated for each SV present in the final measurement block. A B-Value is the difference between the PRC present in the measurement block and the PRC that would exist if a given GPS receiver was excluded. Equation D.16 describes how B-Values are calculated [18].

$$B_{PR}(n,m) = PRC(n) - \frac{1}{M_n - 1} \sum_{i \in S_m;\ i \neq m} PR_{SCA}(n,i) \tag{D.16}$$

where

$PRC(n)$ = Broadcast Pseudorange Correction (meters)

$PR_{SCA}(n,m)$ = Carrier Smoothed & receiver clock adjusted PRC (meters)

S_m = Set of receivers that are tracking the SV

M_n = Number or receivers in set S_m

n = PRN of SV

m = Index of receiver antenna

A measurement block element is only valid if it contains at least two valid B-Values. B-Values are validated only if they satisfy the inequality described in equation D.17 [18]. The pseudorange B-Value threshold (K_{B_PR}) is a configurable value used to validate the B-Values. The K_{B_PR} may only be configured within the range described in equation D.18 [18].

$$B_{PR}(n,m) \leq \frac{K_{B\_PR} \cdot \sigma_{pr\_gnd}(\theta(n))}{\sqrt{M_n - 1}} \tag{D.17}$$

where

K_{B_PR} = Pseudorange B-Value threshold

$$5 \leq K_{B\_PR} \leq 6 \tag{D.18}$$

$\theta(n)$ = Elevation of SV

S_m = Set of receivers that are tracking the SV

M_n = Number or receivers in set S_m

n = PRN of SV

m = Index of receiver antenna

D.2 CL-GBAS Calculations

The corrected position is calculated via a weighted least squares method, but the calculation is simplified to a non-weighted least squares method because $\sigma_{\mathrm{PR\_GND}}$ is a constant value for each SV in LVBS 3.0 [3]. Implementation of a weighted least squares solution is required to perform CL-GBAS calculations when $\sigma_{\mathrm{PR\_GND}}$ varies in the same solution.

D.2.1 Local Monitor Pseudoranges ($\mathbf{PR_{final}}$)

The LM Pseudoranges (PR_{final}) are calculated for each SV that the LM is tracking. The PR_{final} term is calculated by applying the pseudorange smoothing described in section D.1.10.2 to the raw pseudorange measurements from the receiver and then applying corrections to correct for satellite clock errors, group delay, and relativistic effect. Equation D.19 describes the equation used to generate PR_{final} [3].

$$
\begin{aligned}
PR_{final}(n) &= PR_s(n) + t_{sv_{gps}}(n) \qquad \text{(D.19)}\\
&\text{where}\\
PR_{final}(n) &= \text{Local Monitor Pseudorange (seconds)}\\
PR_s(n,m) &= \text{Smoothed Pseudorange (seconds)}\\
t_{sv_{gps}}(n,m) &= \text{Clock Correction (seconds)}\\
n &= \text{PRN of SV}
\end{aligned}
$$

D.2.2 Corrected LM Pseudoranges ($\mathbf{PR_{LMC}}$)

The Corrected LM Pseudorange (PR_{LMC}) is calculated by applying the PRC data present in the Type 1 messages to the PR_{final} term. The PR_{LMC} term is only calculated for SVs that exist in the set of SVs that the local monitor is tracking and those in the set of SVs in the Type 1 message. Equation D.20 describes how PR_{LMC} is calculated [3].

$$
\begin{aligned}
PR_{LMC}(n) &= PR_{final}(n) + PRC(n) \qquad \text{(D.20)}\\
&\text{where}\\
PR_{LMC}(n) &= \text{Corrected LM Pseudorange (seconds)}\\
PR_{final}(n) &= \text{Local Monitor Pseudorange (seconds)}\\
PRC(n) &= \text{Broadcast Pseudorange Correction (seconds)}\\
n &= \text{PRN of SV}
\end{aligned}
$$

D.2.3 Geometry Matrix (G)

Geometry Matrix (G) contains the line of sight (LOS) vectors to all available SVs [1]. G is made up of four columns with one row per SV in the position solution. Equation D.21 describes the construction of G [3].

$$G = \begin{bmatrix} \frac{X_e(n_1)-X_p}{R(n_1)} & \frac{Y_e(n_1)-Y_p}{R(n_1)} & \frac{Z_e(n_1)-Z_p}{R(n_1)} & 1 \\ \frac{X_e(n_2)-X_p}{R(n_2)} & \frac{Y_e(n_2)-Y_p}{R(n_2)} & \frac{Z_e(n_2)-Z_p}{R(n_2)} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{X_e(n_i)-X_p}{R(n_i)} & \frac{Y_e(n_i)-Y_p}{R(n_i)} & \frac{Z_e(n_i)-Z_p}{R(n_i)} & 1 \end{bmatrix} \tag{D.21}$$

where

$(X_e(n), Y_e(n), Z_e(n))$ = ECEF position of SV based on ephemeris

$\left(X_p, Y_p, Z_p\right)$ = ECEF position of receiver directly from the receiver

$R(n)$ = Range between SV and receiver positions (meters)

n = Index value of SV (PRN)

S_i = Set of SVs that are available

i = Index value of S_i

D.2.4 Corrected Position Vector (CPV)

The position correction vector is a vector that can be applied to the ECEF position of the receiver directly from the receiver to generate the Corrected Position Vector (CPV). The position correction vector is generated by solving the non-weighted least squares method according to equation D.22 [3].

$$\langle \Delta x, \Delta y, \Delta z, \Delta b \rangle = \left(G^T \cdot G\right)^{-1} \cdot G^T \cdot \langle \Delta PR \rangle \tag{D.22}$$

where

G = Geometry Matrix

$\langle \Delta PR \rangle = PR_{LMC}(n) - R(n)$; Range Correction Vector $\langle$meters$\rangle$

$PR_{LMC}(n)$ = Corrected LM Pseudorange (seconds)

$R(n)$ = Range between SV and receiver positions (meters)

n = Index value of SV (PRN)

The CPV is generated by adding the position correction vector to the position vector obtained directly from the receiver. Equation D.23 describes how the CPV is generated [3].

$$\langle x_u, y_u, z_u, b_u \rangle = \left\langle x_p, y_p, z_p, 0 \right\rangle + \langle \Delta x, \Delta y, \Delta z, \Delta b \rangle \tag{D.23}$$

where

$\langle x_u, y_u, z_u, b_u \rangle$ = ECEF Corrected Position Vector

$\left\langle x_p, y_p, z_p, 0 \right\rangle$ = ECEF position of receiver directly from the receiver

$\langle \Delta x, \Delta y, \Delta z, \Delta b \rangle$ = Position Correction Vector

D.2.5 Vertical Error (VE) and Horizontal Error (HE)

The Vertical Error (VE) and Horizontal Error (HE) represent the distance in meters that a position solution differs from the surveyed position of the receiver antenna. The CL-GBAS LM calculates the VE and HE based on the surveyed location of the LM receiver antenna and the CPV [3]. LVBS 3.0 calculates the VE and HE that are required for CL-GBAS in addition to VE and HE based on the position available directly from the receiver. The ECEF coordinates are converted to the LLA coordinate system to facilitate the calculation of VE and HE. Equation D.24 describes the calculation used to generate VE [3].

$$VE = Alt_{solution} - Alt_{survey} \tag{D.24}$$

where

VE = Vertical Error (meters)

$Alt_{solution}$ = Altitude from position solution (meters)

Alt_{survey} = Altitude from survey (meters)

The HE is calculated by first calculating the latitude and longitude errors in meters and then calculating the magnitude of the combined error. Equation D.25 describes how latitude error is calculated [3]. Equation D.26 describes how longitude error is calculated [3]. The conversion factors for difference in latitude and longitude from radians to meters are described in equations D.27 and D.28 respectively [3]. Equation D.29 describes how HE is calculated from the latitude and longitude errors [3].

$$Lat_{error} = \left(Lat_{solution} - Lat_{survey}\right) \cdot c_{lat} \tag{D.25}$$

where

Lat_{error} = Latitude Error (meters)

$Lat_{solution}$ = Latitude from position solution (radians)

Lat_{survey} = Latitude from survey (radians)

c_{lat} = Conversion factor for difference in latitude to meters $\left(\frac{\text{meters}}{\text{radian}}\right)$

$$Lon_{error} = \left(Lon_{solution} - Lon_{survey}\right) \cdot c_{lon} \tag{D.26}$$

where

Lon_{error} = Longitude Error (meters)

$Lon_{solution}$ = Longitude from position solution (radians)

Lon_{survey} = Longitude from survey (radians)

c_{lon} = Conversion factor for difference in longitude to meters $\left(\frac{\text{meters}}{\text{radian}}\right)$

$$c_{lat} = \frac{a \cdot \left(1 - \left|1 - \frac{b}{a} \cdot \left(1 + \frac{b}{a}\right)\right|\right)}{\sqrt{1 - \left(\left|1 - \frac{b}{a} \cdot \left(1 + \frac{b}{a}\right)\right| \cdot sin\left(Lat_{survey}\right)^2\right)}^{3}} \tag{D.27}$$

$$c_{lon} = \frac{a \cdot cos\left(Lat_{survey}\right)}{\sqrt{1 - \left(\left|1 - \frac{b}{a} \cdot \left(1 + \frac{b}{a}\right)\right| \cdot sin\left(Lat_{survey}\right)^2\right)}} \tag{D.28}$$

where

c_{lat} = Conversion factor for difference in latitude to meters $\left(\frac{\text{meters}}{\text{radian}}\right)$

c_{lon} = Conversion factor for difference in longitude to meters $\left(\frac{\text{meters}}{\text{radian}}\right)$

Lat_{survey} = Latitude from survey (radians)

a = WGS-84 Equatorial Radius (meters)

b = WGS-84 Polar Radius (meters)

$$HE = \sqrt{(Lon_{error})^2 + (Lat_{error})^2} \tag{D.29}$$

$$\begin{aligned} HE &= \text{Horizontal Error (meters)} \\ Lon_{error} &= \text{Longitude Error (meters)} \\ Lat_{error} &= \text{Latitude Error (meters)} \end{aligned}$$

D.2.6 Vertical Dilution of Precision (VDOP) and Horizontal Dilution of Precision (HDOP)

The Vertical (VDOP) and Horizontal (HDOP) Dilution of Precision terms describe how much a change in GPS measurements affect the GPS position solution. Lower VDOP and HDOP values indicate higher precision in a position solution. Equations D.30 and D.31 represent how VDOP and HDOP are calculated respectively [3].

$$VDOP = G'(3,3) \tag{D.30}$$

$$HDOP = \sqrt{G'(1,1)^2 + G'(2,2)^2} \tag{D.31}$$

where

$$\begin{aligned} VDOP &= \text{Vertical Dilution of Precision} \\ HDOP &= \text{Horizontal Dilution of Precision} \\ G' &= \left(G^T \cdot G\right)^{-1} \\ G &= \text{Geometry Matrix} \end{aligned}$$

D.2.7 Vertical Protection Level (VPL) and Horizontal Protection Level

VPL and HPL are only implemented according to the fault-free (H_0) error hypothesis which describes the error present in the system in the absence of a RS receiver failure. The VPL and HPL are calculated for using the fault-free missed detection multiplier (K_{ffmd}). Table D.7 defines the values used for K_{ffmd} for different GBAS service levels. The non-weighted least squares method used by LVBS 3.0 allows for the simplified implementation of VPL and HPL described in equations D.32 and D.33 respectively.

$$VPL = \sigma_{\text{PR\_GND}} \cdot K_{ffmdV} \cdot VDOP \tag{D.32}$$

$$HPL = \sigma_{\text{PR\_GND}} \cdot K_{ffmdH} \cdot HDOP \tag{D.33}$$

where

$\sigma_{\text{PR\_GND}}$ = Pseudorange Error Bound from Ground Station

K_{ffmdV} = Vertical Fault-Free Missed Detection Multiplier

K_{ffmdH} = Horizontal Fault-Free Missed Detection Multiplier

$VDOP$ = Vertical Dilution of Precision

$HDOP$ = Horizontal Dilution of Precision

Table D.7 Fault-Free Missed Detection Multiplier Values				
	Vertical K_{ffmd}		Horizontal K_{ffmd}	
Number of RS	GSL A, B, C	GSL D, E, F	GSL A, B, C	GSL D, E, F
2	5.762	6.8	5.762	6.9
3	5.810	6.9	5.810	7.0
4	5.847	6.9	5.847	7.0

Table D.7. ***Fault-Free Missed Detection Multiplier Values****: Vertical and horizontal fault-free missed detection multipliers for different GBAS service levels and different numbers of RS in the Type 1 message solutions [29].*

Appendix E - VHF Data Broadcast Equipment Pinouts

Table E.1 VDB J3 Connector

Pins	Description
1	Ground
2-5	Reserved
6	Receiver alarm output
7	Power supply output
8	1PPS RS485 non-inverted input. Logic level 5/0V
9-19	Reserved
20	1PPS RS485 inverted input. Logic level 0/5V
21-22	Reserved
23	Logic ground
24-25	Reserved

Table E.1. ***VDB J3 Connector****: Pinout for RE9009 and EM9009 J3 Remote Control connector, [26] [25].*

Table E.2 VDB J5 Connector

Pins	Description
1	Not used
2	Serial link transmit A. RS485 non-inverted. Logic level 5/0V.
3	Serial link receive A. RS485 non-inverted. Logic level 5/0V.
4-5	Reserved
6	1PPS RS485 non-inverted input. Logic level 0/5V
7	Digital ground
8	Reserved
9	Not used
10-13	Reserved
14	Serial link transmit B. RS485 inverted. Logic level 0/5V.
15	Serial link receive B. RS485 inverted. Logic level 0/5V.
16-17	Reserved
18	1PPS RS485 inverted input. Logic level 0/5V
19-21	Not used

Table E.2 (*cont*)	
22-25	Not used

Table E.2. ***VDB J5 Connector***: *Pinout for RE9009 and EM9009 J5 Data connector, [26] [25].*