4-2019

Algorithm for Geodetic Positioning Based On Angle-Of-Arrival of Automatic Dependent Surveillance-Broadcasts

Richard Allen Gross
Grand Valley State University

ScholarWorks Citation

Gross, Richard Allen, "Algorithm for Geodetic Positioning Based On Angle-Of-Arrival of Automatic Dependent Surveillance-Broadcasts" (2019). *Masters Theses*. 925.
https://scholarworks.gvsu.edu/theses/925

ALGORITHM FOR GEODETIC POSITIONING BASED ON ANGLE-OF-ARRIVAL OF AUTOMATIC DEPENDENT SURVEILLANCE-BROADCASTS

Richard Allen Gross

A Thesis Submitted to the Graduate Faculty of

GRAND VALLEY STATE UNIVERSITY

In

Partial Fulfillment of the Requirements

For the Degree of

Master of Science in Electrical Engineering

Padnos College of Engineering and Computing

April 2019

ABSTRACT

This paper develops a non-precision, three-dimensional, geodetic positioning algorithm for airborne vehicles. The algorithm leverages the proliferation of Automatic Dependent Surveillance – Broadcast (ADS-B) equipped aircraft, utilizing them as airborne navigation aids to generate an RF Angle-of-Arrival (AOA) and Angle-of-Elevation (AOE) based geodetic position. The resulting geodetic position can serve as a redundant navigation system for use during locally limited Global Navigation Satellite System (GNSS) availability, be used to validate on-board satellite navigation systems in an effort to detect local spoofing attempts, and be used to validate ADS-B position reports.

The navigation algorithm is an implementation of an Extended Kalman Filter (EKF) that is loosely based on Simultaneous Localization and Mapping (SLAM), in that it tracks ADS-B capable aircraft while simultaneously determining the geodetic position and velocity of the host vehicle. Unlike SLAM, where the absolute location – latitude/longitude – of the landmarks is unknown and must be estimated as the vehicle encounters them, the absolute position of the airborne navigation aids is typically well-known and periodically reported in the ADS-B data set. Because the absolute position of the navigation aids are known, the resulting host vehicle position will also be an absolute, rather than a relative position. Secondarily, the continuous tracking of the airborne navigation aids allows reported ADS-B positions to be validated against the estimated navigation aid position; thereby, concurrently accomplishing ADS-B validation and host vehicle geolocation.

This research has demonstrated through a series of simulated Monte-Carlo tests that the algorithm is capable of generating valid position estimates, along with a reliable estimate of its accuracy, across a variety of anticipated input conditions. With multiple GNSS quality

navigation aids available, mean position errors below 225 meters were observed. As the quality of the navigation aids decreased, so too did the accuracy of the algorithm. Utilizing navigation aids with an accuracy of 4 nautical miles (95% containment) resulted in mean position errors on the order of 0.75 nautical miles. These results demonstrate that the method is feasible, and even under worst case conditions, the accuracy of the position estimate generated by the algorithm was sufficient to allow an aircraft to navigate to its destination.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

MATHEMATICAL SYMBOLS

Matrices – Typically denoted by boldface capital letters.

$\boldsymbol{A}$	State transition matrix
$\boldsymbol{B}$	Input control matrix
$\boldsymbol{C}$	Coordinate conversion (rotation) matrix
$\boldsymbol{H}$	Measurement matrix
$\boldsymbol{I}$	Identity matrix
$\boldsymbol{K}$	Kalman gain matrix
$\boldsymbol{P}$	State error covariance matrix
$\boldsymbol{Q}$	Process noise covariance matrix
$\boldsymbol{R}$	Measurement noise covariance matrix

Vectors – Typically denoted by boldface lowercase letters.

$\boldsymbol{f}$	Non-linear state transition function
$\boldsymbol{h}$	Non-linear measurement function
$\boldsymbol{m}$	Landmark location vector
$\boldsymbol{r}$	Position vector
$\boldsymbol{u}$	System control input vector
$\boldsymbol{v}$	System measurement (observation) noise vector
$\boldsymbol{w}$	System process noise vector
$\boldsymbol{x}$	System state vector
$\boldsymbol{z}$	System measurement (observation) vector

Sets – Typically denoted by blackboard bold capital letters.

$\mathbb{M}$	Complete set of all landmarks (landmark set)
$\mathbb{U}$	Complete history of control inputs (input set)
$\mathbb{X}$	Complete history of vehicle location (location set)
$\mathbb{Z}$	Complete set of all observations (observation set)

Variables – Typically denoted by normal face characters.

α	Angle of arrival
β	Angle of elevation
e	Earth eccentricity
f	Earth flattening
φ	Geocentric latitude
Φ	Geodetic latitude
λ	Longitude
Ψ	Azimuth angle
s	Spherical distance
θ	Geocentric angle
R_0	Equatorial radius of the Earth
R_E	Transverse radius of curvature
R_m	Mean radius of the Earth
R_N	Meridian radius of curvature
R_p	Polar radius of the Earth

Accents

Circumflex, ˆ	Estimated value
Diacritic, ˙	First derivative
Diaresis, ¨	Second derivative
Tilde, ˜	Measured value

ACRONYMS AND ABBREVIATIONS

ADS-B	Automatic Dependent Surveillance Broadcast
AOA	Angle of Arrival
AOE	Angle of Elevation
AOI	Angle of Intersection
API	Application Program Interface
CDF	Cumulative Distribution Function
DF	Direction Finding
DGPS	Differential Global Positioning System
DOP	Dilution of Precision
ECEF	Earth Centered, Earth Fixed
EKF	Extended Kalman Filter
ENU	East, North, Up Frame
EPU	Estimated Position Uncertainty
ES	Extended Squitter
ESPRIT	Estimation of Signal Parameters via Rotational Invariance Techniques
FAA	Federal Aviation Administration
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GLONAS	Globalnaya Navigazionnaya Sputnikovaya Sistema
GVA	Geometric Vertical Accuracy
HIL	Horizontal Integrity Limit
HPL	Horizontal Protection Limit

ICAO	International Civil Aviation Organization
INS	Inertial Navigation System
LOS	Line of Sight
MUSIC	Multiple Signal Classification
NACp	Navigation Accuracy Category for Position
NACv	Navigation Accuracy Category for Velocity
NC ESPRIT	Non-Circular Unitary ESPRIT
NED	North, East, Down Frame
NIC	Navigation Integrity Category
MPU	Measured Position Uncertainty
PDF	Probability Density Function
R_C	Radius of Containment
RF	Radio Frequency
RMSE	Root Mean Square Error
RPE	Radial Position Error
RTCA	Radio Technical Commission for Aeronautics
RTK	Real Time Kinematics
SVD ESPRIT	Single Value Decomposition ESPRIT
SLAM	Simultaneous Localization and Mapping
USB	Universal Serial Bus
VEPU	Vertical Estimated Position Uncertainty
VMPU	Vertical Measured Position Uncertainty
VPE	Vertical Position Error

WAAS	Wide Area Augmentation System
WGS-84	World Geodetic System 1984

1.0 INTRODUCTION

1.1 Problem Description

Since their deployment, Global Navigation Satellite Systems (GNSS) have set the standard for geodetic positioning. These satellite based navigation systems include the likes of the Global Positioning System (GPS) developed by the United States, the Galileo system deployed by the European Union, and the Globalnaya Navigazionnaya Sputnikovaya Sistema (GLONAS) developed by Russia. Unaided, the highly engineered nature of these systems can produce geodetic position estimates with errors on the order of 10 meters. Various augmentation methods have been developed to further improve the accuracy of GNSS, including but not limited to: Wide Area Augmentation System (WAAS), Differential Global Positioning System (DGPS), and Real Time Kinematic (RTK) GNSS. Utilizing these augmentation methods can improve the accuracy of the GNSS position estimate to the centimeter range. Despite the unparalleled accuracy of GNSS systems, the low power of the satellite-based Radio Frequency (RF) signals required to formulate the geodetic position estimates leaves GNSS susceptible to a lack of availability and spoofing. GNSS may become unavailable due to natural phenomena, obstructions in the line of sight to the satellite constellation, or malicious intent. Spoofing, on the other hand, is the intentional introduction of a higher power 'look-a-like' GNSS signal that causes the GNSS receiver to report an incorrect position estimate. It has been widely theorized that spoofing can be used to 'take control' of a GNSS guided vehicle; therefore, significant research has been performed to detect spoofing attempts.

This paper develops a non-precision, three-dimensional, geodetic positioning algorithm for airborne vehicles that is capable of determining the location of the host airborne vehicle with position errors on the order of 225 meters. The algorithm leverages the proliferation of Automatic Dependent Surveillance – Broadcast (ADS-B) equipped aircraft by utilizing them as airborne navigation aids. The algorithm exploits available ADS-B transmission to generate an RF angle-of-arrival (AOA) and angle-of-elevation (AOE) based geodetic position estimate for the aircraft receiving the ADS-B data (the host vehicle). The resulting host vehicle position data can serve as a redundant navigation system for use during locally limited GNSS availability, be used to validate on-board satellite navigation systems in an effort to detect local spoofing attempts, and be used to validate ADS-B position reports from other aircraft.

The algorithm requires that a minimum of two ADS-B capable aircraft are operating within ADS-B range of the host vehicle, and that the AOA from these two aircraft form an angle-of-intersection (AOI) with the host vehicle that is in the range [20, 60] degrees. An illustration depicting the concept is provided in Figure 1-1 with additional details to follow.

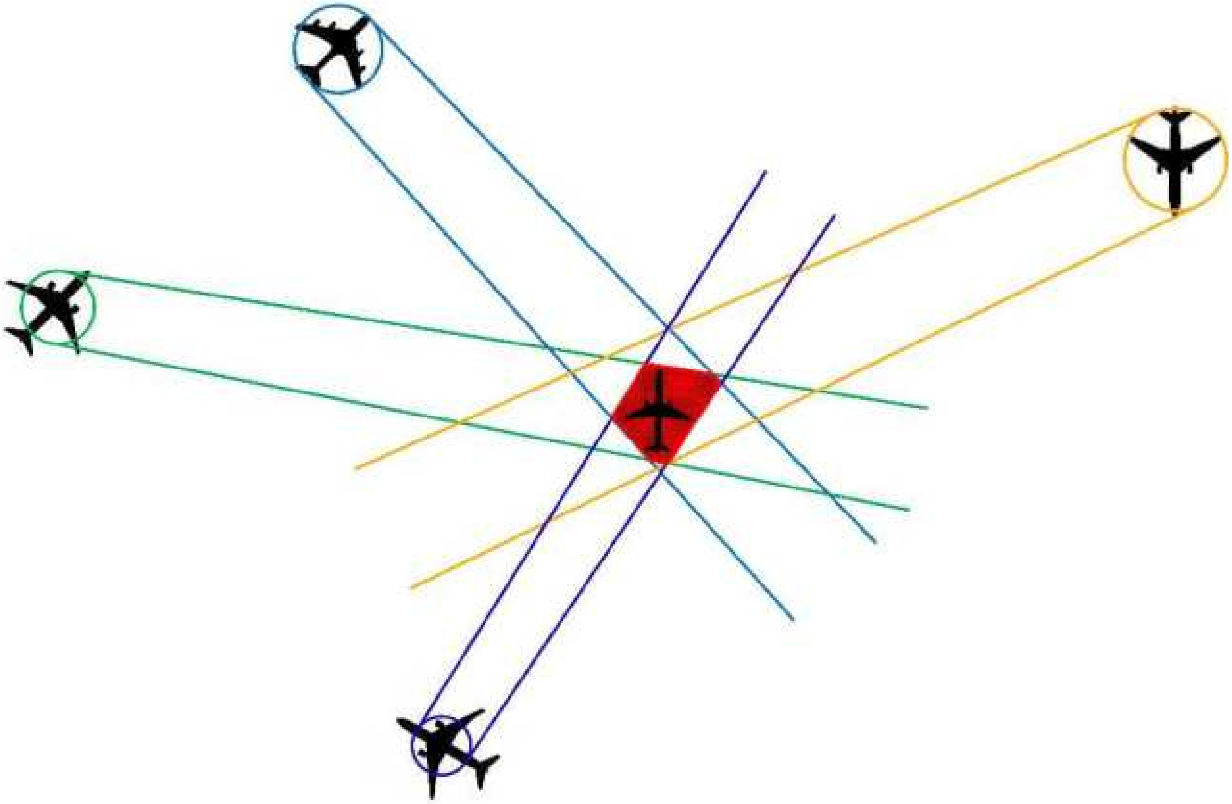

Figure 1-1: Illustration of the Fundamental Concept to Determine the Geodetic Position of an Aircraft Based on the Angle-of-Arrival of ADS-B Data from Aircraft Operating in the Vicinity.

Each airborne navigation aid (ADS-B capable aircraft around the perimeter of Figure 1-1) periodically transmits its position and velocity, along with an indication of the associated uncertainty in these estimates. The magnitude of the position uncertainty for a particular ADS-B aircraft is denoted by the size of the circle that is centered on each airborne navigation aid. The receiving aircraft measures the AOA and AOE of the ADS-B data, and associates the AOA and AOE measurements with the position information from the transmitting aircraft. Combining this position, AOA, and AOE data from multiple aircraft – and accounting for the uncertainty in the transmitting aircraft's position, the AOA uncertainty, and the AOE uncertainty – results in a closed polygonal region of uncertainty in which the receiving aircraft must reside. This region of uncertainty is depicted by the red cross-hatched polygon in Figure 1-1.

Although this basic concept is easily envisioned as a deterministic process, the theory is complicated by several limitations. First, all of the aircraft are in continuous motion. Second, ADS-B position reports are subject to latency of up to 0.6 seconds [1]. Third, there is uncertainty in the AOA and AOE measurements. And finally, ADS-B position reports from a particular aircraft are transmitted at a staggered interval [1], minimizing the likelihood of simultaneously receiving position reports from multiple airborne navigation aids. Due to these constraints, the receiving aircraft position is not calculated based on the simple deterministic triangulation technique illustrated in Figure 1-1. Instead, the system is treated as a stochastic process, whose system state is estimated using an implementation of the Extended Kalman Filter (EKF).

The navigation algorithm itself is loosely based on Simultaneous Localization and Mapping (SLAM), in that it tracks ADS-B capable aircraft while simultaneously determining the geodetic position and velocity of the host vehicle. Unlike SLAM, where the absolute location – latitude/longitude – of the landmarks is unknown and must be estimated as the vehicle encounters them, the absolute position of the airborne navigation aids is reasonably well-known and periodically reported in the ADS-B data set. Because the absolute position of the navigation aids is known, the resulting host vehicle position will also be an absolute, rather than a relative position. Secondarily, the continuous tracking of the airborne navigation aids allows reported ADS-B positions to be validated against the estimated navigation aid position; thereby, concurrently accomplishing ADS-B validation and host vehicle geolocation. Finally, unlike GNSS systems that utilize low power RF signals, ADS-B transmissions are relatively high power, ranging from 70 W to 200 W [1], making them very difficult to jam or spoof.

In summary, this paper presents a non-precision means of determining an airborne vehicle's geodetic position, even in the absence of GNSS position data. Monte-Carlo analysis of the algorithm indicates that position errors less than 225 meters can be achieved when multiple ADS-B capable aircraft are utilized to construct the position estimate. In addition, this algorithm can be used to validate GNSS position reports in a GNSS spoofed environment, and can serve as a means to validate ADS-B position reports from suitably equipped aircraft operating in proximity to the host vehicle.

1.2 Scope

The purpose of this research is to investigate an alternative geodetic positioning technique to serve as a non-precision alternative/supplement to GNSS. The system will calculate the geodetic position of the host vehicle based on the AOA and AOE of received ADS-B signals from aircraft operating within the radio horizon of the host vehicle. The resulting geodetic position could then be used to validate GNSS measurements, serve as a failsafe alternative in the event of total GNSS failure, or be used to validate ADS-B position reports.

It must be emphasized that the current investigation is theoretical in nature; implying that the development of custom hardware or the implementation of a fully operational system falls outside of the scope of this research.

1.3 Assumptions

It is assumed that accurately determining the AOA of ADS-B signals is feasible based on the research conducted by Reck et al. in [2], [3], [4], [5], [6], [7] and by Faragher in [8]. Both of these researchers, whose publications are summarized in Section 3.0 – Review of Literature, empirically demonstrated that the AOA of ADS-B data can be determined at a stationary antenna with a Root Mean Square Error (RMSE) of less than 1 degree. The assumption is made for this research that the AOE of ADS-B signals can be determined with similar accuracy as AOA. Several companies produce 3-dimensional Direction Finding (DF) antennas capable of providing both AOA and AOE information including: Applied Signals Intelligence [9] and Aaronia [10]; however, neither of these companies makes AOE/AOE accuracy data publicly available.

The research further assumes that the host vehicle contains a sensor (or suite of sensors) capable of providing vehicle orientation and true heading information. This will allow the received AOA and AOE data to be expressed in a local coordinate frame relative to true north. It is assumed that it will be a function of the direction finding receiver to perform the necessary rotations to express the AOA and AOE in a locally level frame. Practically, this could be performed by either the direction finding receiver or the by the navigation algorithm. However, for clarity of expressing the navigation algorithm, the AOA/AOE rotations will not be included in the navigation equations.

Finally, it is assumed that utilizing a spherical Earth model in place of the more complex and realistic oblate spheroid model will provide sufficient accuracy for the purposes of this analysis. The level of relative error due to the Earth's flattening over the somewhat small distance to the radio horizon is expected to result in a small absolute errors that can simply be treated as an additional source of uncertainty in the implementation.

2.0 BACKGROUND

This research develops an approach to geodetic positioning that includes the use of a somewhat new and emerging technology. Most engineering aspects of the algorithm are well understood, but have not been combined in the manner presented in this work. As such, a somewhat thorough background is developed in the areas of: ADS-B, Kalman filtering, SLAM, Gaussian and Rayleigh random variables, and the equations of geodesic paths and positioning.

2.1 Automatic Dependent Surveillance – Broadcast

ADS-B is a modern technology that has been designed to enhance air traffic control's situational awareness of aircraft operations by providing a three-dimensional depiction of each ADS-B equipped aircraft's intended flight path [11]. To accomplish this, each ADS-B equipped aircraft periodically transmits its identification, position, altitude, velocity, and other aircraft specific information [11]. Both ground stations and suitably equipped aircraft may receive these transmissions, enhancing air traffic control and allowing aircraft to maintain adequate separation in both controlled and uncontrolled airspace [12].

The Federal Aviation Administration (FAA) has mandated that all aircraft operating within certain airspace segments over the United States be ADS-B compliant by January 1, 2020 [12]. The airspace segments requiring ADS-B capability are depicted in Figure 2-1, where the required airspace is denoted by the terms 'UAT or 1090ES' or '1090ES Required'.

Although not yet regulatory standard, many aircraft owners have opted to be early adapters of the technology. To be ADS-B compliant, each aircraft is required to transmit ADS-B data packets on either 978MHz or 1090MHz in accordance with AC 90-114A [12]. All aircraft operating in Class A airspace are required to utilize the 1090MHz frequency instead of 978MHz [12]. Class A airspace, which extends upwards from 18,000ft mean sea level, is under air traffic control, and is typically utilized by commercial aircraft on cross-country flights.

According to 2015 data released by the Bureau of Transportation Statistics, there are over 6,800 commercial aircraft and 210,000 general aviation aircraft registered in the United States [13]. A vast majority of these aircraft will be subject to the ADS-B mandate, making them ideal candidates to serve as airborne navigation aids in the navigation system described in this paper.

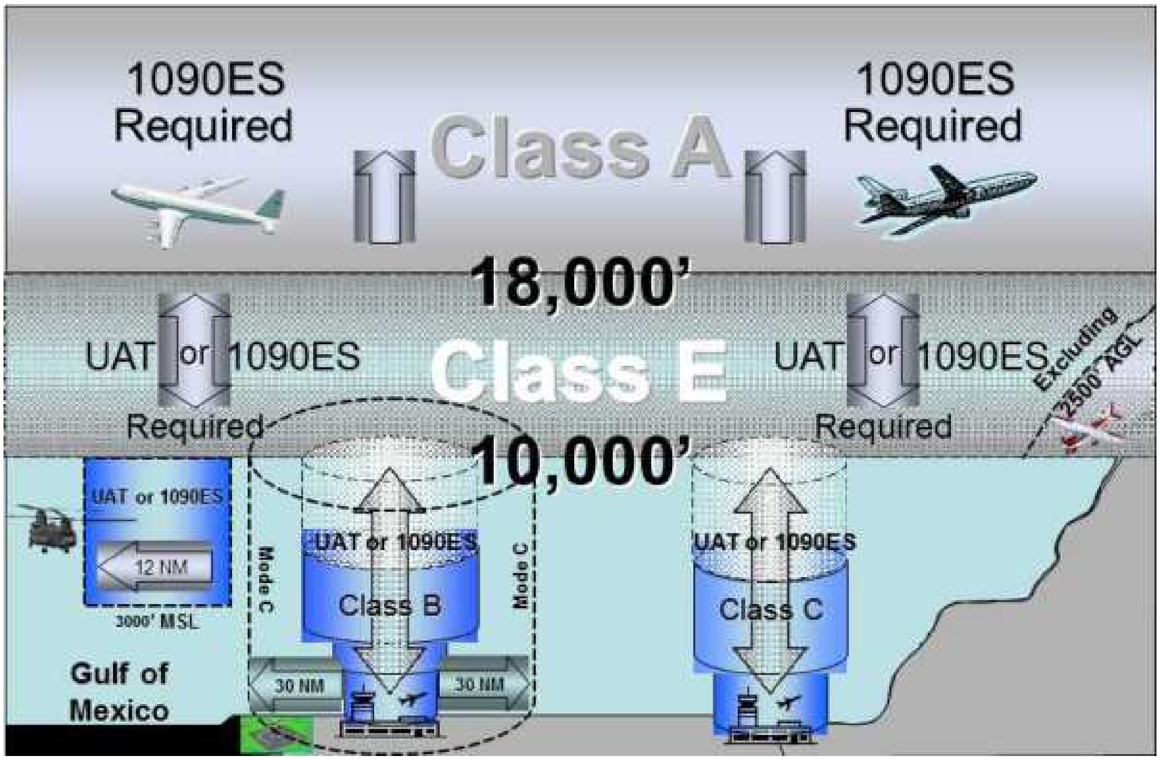

Figure 2-1: Airspace Segments Requiring ADS-B Compliance [12].

2.1.1 ***Data Payload***

The ADS-B data transmitted on 1090MHz is an extension of aircraft transponder Mode S capability called Extended Squitter (ES) [1]. The extended data is transmitted in spontaneous broadcasts that are 120µs in duration, containing 120 bits, each with a duration of 0.5µs [1]. The initial 8 bits contain the message preamble, while the remaining 112 bits contain the message payload [1]. The 112 bit payload is divided into the five fields shown in Table 2-1 [1].

Table 2-1: 1090 MHz Extended Squitter Payload [1].

LSB	MSB	Payload Abbreviation	Payload Nomenclature
1	5	DF	Downlink Format
6	8	CA	Capability (Message Subtype)
9	32	AA	ICAO Aircraft Address
33	88	ME	Message Data (ADS-B Data)
89	112	PI	Parity/Interrogator ID

The 5-bit Downlink Format field and the 3-bit Capability field provide information describing the type transmitter used to source the ADS-B data and its unique capabilities [1]. This information is not directly applicable to the current research and is thus excluded from further description. Likewise, the 24-bit Parity/Interrogator ID is also of little consequence to this research and additional detail has not been provided.

2.1.1.1 ICAO Aircraft Address Field

The International Civil Aviation Organization (ICAO) Aircraft Address field contains the unique ICAO identifier for the transmitting installation [1]. This 24-bit field allows for unambiguous identification of the source of the transmitted ADS-B data.

2.1.1.2 Message Data Field

The Message Data field is comprised of 56-bits, constituting the bulk of the pertinent ADS-B data [1]. The first 5-bits of the Message Data field contain the Type Code, which identifies the content of the data contained within the remaining 51-bits [1]. Of particular interest to this research are the Identification, Airborne Position, Airborne Velocity, and Aircraft Operational Status message types given in Table 2-2.

Table 2-2: Subset of 1090MHz Extended Squitter Message Formats [1].

Type Code	Description	Altitude Type	Navigation Integrity Category (NIC)
1	Identification (Category Set D)	N/A	N/A
2	Identification (Category Set C)	N/A	N/A
3	Identification (Category Set B)	N/A	N/A
4	Identification (Category Set A)	N/A	N/A
9	Airborne Position	Barometric	11
10	Airborne Position	Barometric	10
11	Airborne Position	Barometric	9,8
12	Airborne Position	Barometric	7
13	Airborne Position	Barometric	6
14	Airborne Position	Barometric	5
15	Airborne Position	Barometric	4
16	Airborne Position	Barometric	3,2
17	Airborne Position	Barometric	1
18	Airborne Position	Barometric	0
19	Airborne Velocity	Either	N/A
20	Airborne Position	GNSS	11
21	Airborne Position	GNSS	10
22	Airborne Position	GNSS	0
31	Aircraft Operational Status	N/A	N/A

Note: Type codes 5-8 and 23-30 were intentionally omitted because they do not provide information relevant to this application.

2.1.1.2.1 Aircraft Identification Messages

As indicated in Table 2-2, four separate Type Codes define Aircraft Identification message Category Sets A, B, C, and D. Combining the Category Set with the 3-bit Emitter Category subfield (bits 6-8 of the Aircraft Identification Message) provides insight into the type of vehicle upon which the ADS-B emitter is affixed. The various Category Set/Emitter Category combinations are described in Table 2-3 [1]. Note that Category Set D is reserved for future use; as such, no specific details are provided.

The remaining 48-bits in the Aircraft Identification Message provide the flight number, aircraft registration number, or other identifying character strings specific to the vehicle upon which the emitter is installed [1].

Table 2-3: ADS-B Emitter Category Descriptions [1].

Value	Category Set A Interpretation	Category Set B Interpretation	Category Set C Interpretation
0	No ADS-B Emitter Category Information		
1	Light (<15500 lbs.)	Glider	Surface Vehicle (Emergency)
2	Small (15500 to 75000 lbs.)	Lighter-than-Air	Surface Vehicle (Service)
3	Large (75000 to 300000 lbs.)	Parachutist	Point Obstacle
4	High-Vortex Large Aircraft	Ultralight/Hang-Glider	Cluster Obstacle
5	Heavy (> 300000 lbs.)	Reserved	Line Obstacle
6	High Performance	Unmanned Aerial Vehicle	Reserved
7	Rotorcraft	Space Vehicle	Reserved

2.1.1.2.2 Airborne Position Messages

As indicated in Table 2-2, thirteen different Type Codes define a variety of Airborne Position messages. The Airborne Position messages provide the transmitting aircraft's altitude, latitude, and longitude expressed with respect to the World Geodetic System 1984 (WGS-84) ellipsoid [1]. The Type Code indicates the Radius of Containment (R_C) of the horizontal navigation error, as well as the source of the altitude data as either barometric or GNSS [1]. A cross reference of the Navigation Integrity Category (NIC) expressed in Table 2-2 to the Radius of Containment is given in Table 2-4. DO-260B does not specifically define the percent probability of containment associated with this R_C value; however, DO-260B does indicate that GNSS Horizontal Protection Limit (HPL) or Horizontal Integrity Limit (HIL) are acceptable sources for the R_C value. HPL and HIL values are typically assumed to constitute a 99.999% radius of containment, but this is not explicitly stated in DO-260B.

Table 2-4: Navigation Integrity Category to Radius of Containment Value [1].

NIC	R_C	NIC	R_C
0	$R_C \geq 20$nmi	6	$0.2 \leq R_C < 0.5$nmi
1	$8 \leq R_C < 20$nmi	7	$0.1 \leq R_C < 0.2$nmi
2	$4 \leq R_C < 8$nmi	8	$75\text{m} \leq R_C < 0.1$nmi
3	$2 \leq R_C < 4$nmi	9	$25 \leq R_C < 75$m
4	$1 \leq R_C < 2$nmi	10	$7.5 \leq R_C < 25$m
5	$0.5 \leq R_C < 1$nmi	11	$R_C < 7.5$m

It must be noted that the vehicle position information transmitted in the Airborne Position message is subject to latency. AC 20-165B indicates that the transmitting aircraft must transmit its position within 2.0 seconds of measurement [11]. Additionally, AC 20-165B indicates that the transmitted position must be latent by less than 0.6 seconds, and may not be overcompensated by more than 0.2 seconds [11]. This implies that if the avionics system in the transmitting aircraft cannot transmit its measured position within 0.6 seconds of calculation, then it must extrapolate its position forward in time so that the reported position is within the window of +0.2/-0.6 seconds of the time of transmission. The Radio Technical Commission for Aeronautics (RTCA) recommends a latency of less than 0.4 seconds to support future ADS-B applications [11].

2.1.1.2.3 Airborne Velocity Message

The Airborne Velocity message describes the transmitting aircraft's velocity over the ground in a Cartesian north/east frame, the vertical rate, and a measure of the 95% probability horizontal velocity error. The horizontal velocity error is referred to as Navigation Accuracy Category for Velocity (NACv) and has the interpretation given in Table 2-5.

Table 2-5: Navigation Accuracy Category for Velocity [1].

NACv	Horizontal Velocity Uncertainty (95%)
0	Unknown or ≥ 10 m/s
1	< 10 m/s
2	< 3.0 m/s
3	< 1.0 m/s
4	< 0.3 m/s

2.1.1.2.4 Aircraft Operational Status Message

The Aircraft Operational Status message provides information related to the current status of the aircraft [1]. Of interest to this application are the Navigation Accuracy Category for Position (NACp) and the Geometric Vertical Accuracy (GVA) fields.

The NACp field provides a measure of the aircraft's Estimated Position Uncertainty (EPU) expressed as a 95% radial value [1]. That is, EPU defines the radius of a circle, centered on the reported position, where the probability of the actual position lying inside the circle is 95% [1]. Unlike the R_C provided in the Airborne Position Message, the NACp is clearly defined as a 95th percentile radius of containment. Table 2-6 provides the relationship between NACp and EPU.

Table 2-6: Navigation Accuracy Category for Position [1].

NACp	EPU (95%)	NACp	EPU (95%)
0	EPU ≥ 10 NM	6	EPU < 0.3 NM
1	EPU < 10 NM	7	EPU < 0.1 NM
2	EPU < 4 NM	8	EPU < 0.05NM
3	EPU < 2 NM	9	EPU < 30 m
4	EPU < 1 NM	10	EPU < 10 m
5	EPU < 0.5 NM	11	EPU < 3 m

Similarly, the GVA field describes the one-dimensional, 95% vertical uncertainty as described in Table 2-7.

Table 2-7: Geometric Vertical Accuracy [1].

GVA	Vertical Uncertainty (95%)
0	Unknown or > 150 m
1	≤ 150 m
2	≤ 45 m
3	Reserved

2.1.2 Transmit Power

The minimum RF peak output power requirements for ADS-B transmissions are a function of the transponder class. A minimum RF peak power of 70 W is required for Class A0 and Class B transponders. These low power transponders are installed on ground vehicles and aircraft subject to visual flight rules [14]. Class A1 and A2 transponders, used on aircraft subject to instrument flight rules, require a minimum peak power of 125 W [1] [14]. Finally, Class A3 transponders, used on extended capability aircraft operating in Class A airspace, require a minimum peak power of 200 W [1]. A maximum peak output power of 500 W is specified for all transponder classes [1].

2.1.3 Transmission Rates

Each ADS-B transmitter operating on the network emits spontaneous broadcasts limited to a maximum of 6.2 messages per second, as averaged over a 60 second interval [1]. The transmission of each individual message from the ADS-B message set is staggered and broadcast at a uniformly distributed random interval [1]. The sparseness of transmissions, along with the random/staggered interval, reduces the number of data collisions experienced at the receiver, and prevents multiple aircraft from continuously masking one another due to unintended transmission synchronization. The uniform distribution parameters for the messages of interest to this application are given in Table 2-8 [1].

Table 2-8: Uniformly Distributed ADS-B Message Transmission Intervals [1].

ADS-B Message	Minimum Interval [s]	Maximum Interval [s]
Airborne Position	0.4	0.6
Airborne Velocity	0.4	0.6
Operational Status	2.4	2.6

2.1.4 Receiver Sensitivity

Like the transmission power described in Section 2.1.2, receiver sensitivity is also a function of the transponder class. Class A0 receivers must be capable of detecting and decoding ADS-B packets received at -72 dBm [1]. Class A1 and A2 receivers utilize a -79 dBm threshold, while Class A3 receivers employ an -84 dBm threshold [1].

2.1.5 Minimum Desired Range

In addition to specifying transmitter power and receiver sensitivity requirements for ADS-B, DO-260B also specifies minimum range requirements in which a receiving system is expected to reliably receive data from a transmitting aircraft. The minimum desired ranges as a function of equipment category are given in Table 2-9.

Table 2-9: Desired Minimum Range for ADS-B Systems [1].

Equipment Category	Desired Minimum Range [NM]
Class A0	10
Class A1	20
Class A2	40
Class A3	120

2.1.6 Theoretical ADS-B Range

The ADS-B ranges given in Section 2.1.5 represent the minimum desired range for ADS-B installations. Based on both free space propagation and line-of-sight analysis, ADS-B data will theoretically be available beyond these desired operational minimums. This section presents an analysis for both free space and line-of-sight propagation to derive the theoretical ADS-B operational range.

2.1.6.1 Theoretical Free Space Propagation

Assuming RF signal transmissions under the effects of free space propagation, the RF signal power at a receiver is given by (2-1) [14].

$$P_R = P_T G \left(\frac{\lambda}{4}\pi R\right)^2 \tag{2-1}$$

Where: P_R = Minimum receiver power.
P_T = Transmitter power.
G = Antenna gain.
λ = Wavelength.
R = Radial distance in meters.

Solving (2-1) for the radial distance leads to (2-2) [14].

$$R = \sqrt{\frac{P_T}{P_r}}\left(\frac{G\lambda}{4\pi}\right) \tag{2-2}$$

Assuming an average transmit power of one-half of the peak power described in Section 2.1.2, a receiver sensitivity of -90dBm (1pW), and an antenna gain of 2.1dBi (1.622) [14], the theoretical ADS-B range for a Class A0 transmitter is established in (2-3). Subject to these assumptions, the theoretical range for each transmitter class is given in Table 2-10. It should be noted that the assumed gain of 2.1dBi is typical of blade style airborne transponder antennas. Although the radiation pattern for these antennas is not purely omnidirectional, this research has assumed it to be so. This research further assumes that the multiple element antenna arrays required for direction finding will have similar gain.

$$R = \sqrt{\frac{35}{1x10^{-12}}\left[\frac{1.622(0.275)}{4\pi}\right]} = 210\ km = 113.5\ NM \quad \textbf{(2-3)}$$

Table 2-10: Theoretical ADS-B Range Based on Free Space Propagation.

Transmitter Class	Peak Power [W]	Average Power [W]	Range [km]	Range [NM]
A0	70	35	210	115
A1 & A2	125	75	300	165
A3	200	100	355	190

2.1.6.2 Line of Sight

When operating near the surface of the Earth, the theoretical transmission range established through analysis of free space propagation could be limited by the Earth itself. That is, the Earth may interfere with direct RF propagation. To ensure that direct RF propagation is achievable, the transmitter must have line-of-sight to the receiver. As shown in Figure 2-2, the maximum direct line-of-sight distance is achieved when the line connecting the transmitter (P_1) and the receiver (P_2) is tangential to the surface of the Earth.

Based on the geometry depicted in Figure 2-2, the maximum line-of-sight distance as a function of transmitter altitude (h_1) and receiver altitude (h_2) can be derived using the Pythagorean Theorem. This is shown in (2-4) through (2-7).

$$d_1^2 = (R_m + h_1)^2 - R_m^2 = 2R_m h_1 + h_1^2 \quad \textbf{(2-4)}$$

Where: R_m= Mean radius of the Earth.

$$d_2^2 = (R_m + h_2)^2 - R_m^2 = 2R_m h_2 + h_2^2 \quad \textbf{(2-5)}$$

$$LOS = d_1 + d_2 = \sqrt{2R_m h_1 + h_1^2} + \sqrt{2R_m h_2 + h_2^2} \quad \textbf{(2-6)}$$

Applying the simplifying assumption of $h_i^2 \ll 2R_m h_i \therefore h_1 = h_2 \rightarrow 0$ to (2-6) results in an expression for line-of-sight distance based on aircraft altitudes. This expression is given as (2-7).

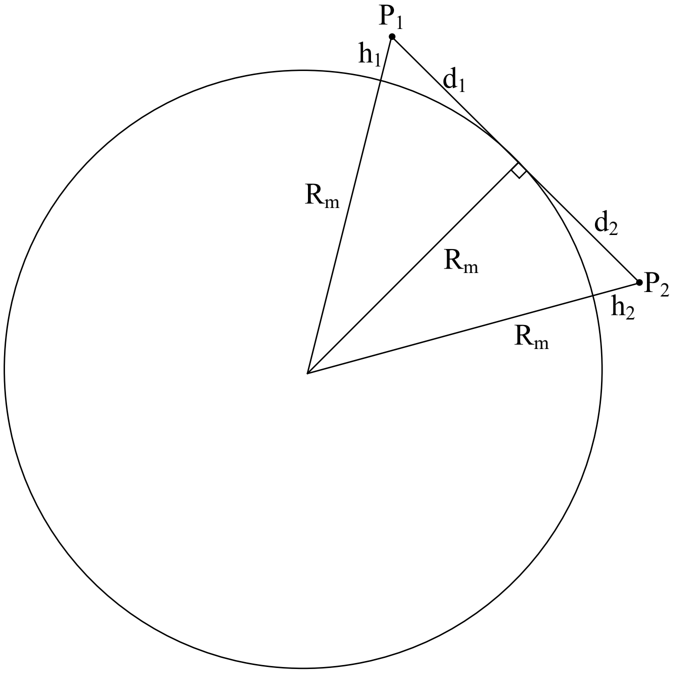

Figure 2-2: Geometry for Line Of Sight Derivation.

$$LOS = \sqrt{2R_m h_1} + \sqrt{2R_m h_2} = \sqrt{2R_m}\left(\sqrt{h_1} + \sqrt{h_2}\right) \quad \textbf{(2-7)}$$

2.1.6.3 <u>Assumed ADS-B Range</u>

Figure 2-3 provides a comparison of the theoretical free space propagation range and the line of sight distance. The three horizontal black traces indicate the free space propagation range for the three transmitter power levels. Likewise, the ensemble of non-linear traces show the line-of-sight range as a function of transmitter and receiver altitude.

This graph demonstrates that line-of-sight is the limiting factor at low altitudes, while free space propagation limits the theoretical ADS-B range at higher altitudes. Given this observation, the theoretical ADS-B range for purposes of this research is assumed to be the minimum of the free space propagation distance and the line-of-sight distance.

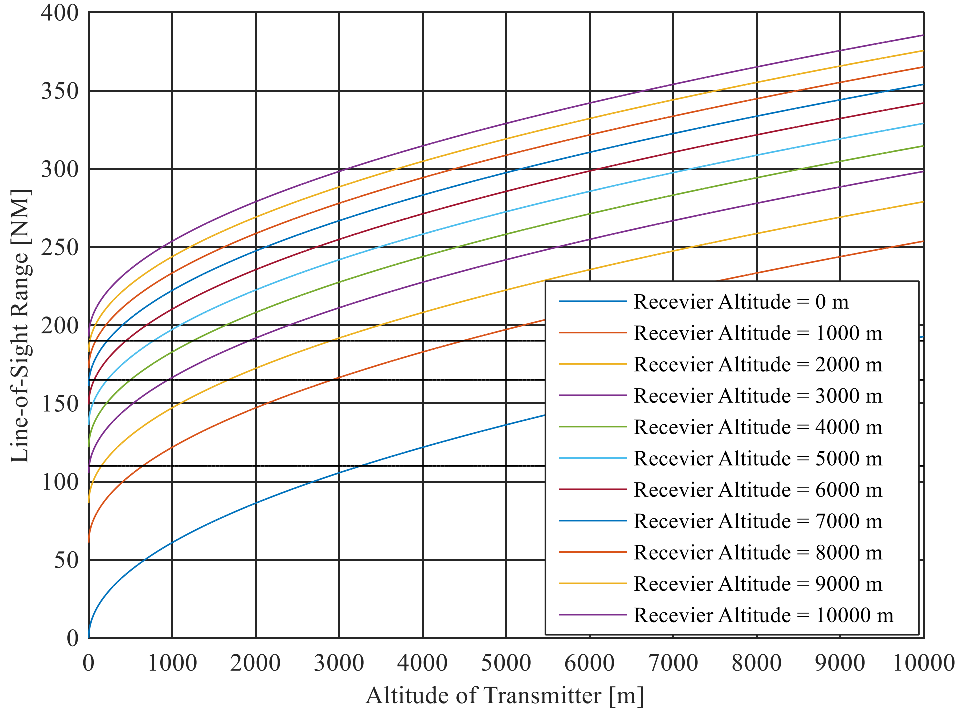

Figure 2-3: Line-of-Sight as a Function of Transmitter and Receiver Altitude.

2.1.7 Nominal ADS-B Reported Accuracy

To gauge the nominal reported accuracy of ADS-B transmissions for use in evaluation of this algorithm, a survey of the NACp, NACv, and GVA values from 2,417 unique ADS-B capable aircraft was conducted. The hardware consisted of a NooElec USB RTLSDR ADS-B receiver, while the software was a locally modified version of the open source Dump1090 Mode S decoder. The open source software was modified to decode and record the ICAO, NACp, NACv, and GVA values for unique aircraft that were transmitting non-zero accuracy values. Once recorded and saved, the data was post processed to determine the mean, median, maximum, and minimum NACp, NACv, and GVA values. The survey results are presented as Table 2-11.

Table 2-11: Reported ADS-B Accuracy Statistics.

Parameter	Mean	Median	Maximum	Minimum
NACp	9.66	10	11	6
NACv	1.48	1	4	1
GVA	1.98	2	2	1

2.2 Radio Frequency Angle of Arrival/Direction Finding

Angle-of-Arrival, also known as direction finding, is a method of determining the direction from which an RF wave originated. Radio navigation, radar, sonar, pseudo-doppler, and seismic exploration are but a handful of the applications that rely on high-resolution AOA estimation [15]. Given the wide variety of applications dependent on AOA estimation, it should come as no surprise that it has been the subject of significant research.

AOA estimation can be thought of as a far-field plane wave intersecting a uniform linear array of antennas [6]. Based on the impinging angle, the signals received at adjacent antennas will demonstrate a characteristic phase offset [6]. This phase relationship can then be used to estimate the AOA of the received signal [6].

Early approaches to the AOA estimation problem included Burg's (1967) maximum entropy method and Capon's (1969) maximum likelihood method [15]. Although widely used, these approaches were fundamentally flawed because their models failed to account for additive noise present in the received signal [15]. More modern approaches that employ a more robust model to account for this noise include: Multiple Signal Classification (MUSIC) [16], Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) [15], Non-Circular Unitary ESPRIT (NC ESPRIT) [17], Single Value Decomposition ESPRIT (SVD ESPRIT) [18], and Matrix Pencil [19].

Due to the fact that the algorithm described in this paper is simply a consumer of the AOA data, and no effort is being made to construct an AOA ADS-B receiver, a detailed exploration of these methods is not provided. The interested reader is encouraged to review the referenced documents for more details.

2.3 Kalman Filter

The Kalman filter was developed in 1960 by Rudolph E. Kalman and continues to be widely used in the areas of navigation, ballistic missile trajectory estimation, and radar/fire control [20] [21]. A Kalman filter is an optimal, linear, least squared, estimator that can provide valid estimations of the current and future state of a linear dynamic system, even when the exact dynamics of the modeled system are unknown [20]. That is, the filter provides an efficient means to estimate the state and covariance of a linear dynamic system that is subject to white noise by processing measurements that are also subject to white noise [22].

The symbolic notation defined in Table 2-12 is provided for reference and to aid in the description of the equations presented in the remainder of this section. Unless otherwise specified, vectors are defined to be column vectors.

Table 2-12: Notation for Kalman Filter Parameters [22].

Symbol	Vector Name	Size	Symbol	Matrix Name	Dim
k	Discrete time index	1	$\boldsymbol{A}$	State transition matrix	n x n
$\boldsymbol{x}$	System state	n	$\boldsymbol{P}$	State error covariance matrix	n x n
$\hat{\boldsymbol{x}}$	Estimate of state	n	$\boldsymbol{B}$	Input control matrix	n x r
$\boldsymbol{u}$	Control input	r	$\boldsymbol{Q}$	Process noise covariance matrix	r x r
$\boldsymbol{w}$	Process noise	r	$\boldsymbol{H}$	Measurement geometry matrix	l x n
$\tilde{\boldsymbol{z}}$	Measurement	l	$\boldsymbol{R}$	Measurement noise covariance matrix	l x l
$\boldsymbol{v}$	Measurement noise	l	$\boldsymbol{K}$	Kalman Gain matrix	n x l

As previously noted, the Kalman filter seeks to estimate the state and covariance of a discrete time process defined by the well-known linear stochastic difference equation given in (2-8) [20] [21] [22] [23] [24]. The state estimates are produced through observation of a measurement that takes on the form given in (2-9) [20] [21] [22] [23] [24]. The variables $\boldsymbol{w}_k$ (process noise) and $\boldsymbol{v}_k$ (measurement noise) are assumed to be normally distributed, zero mean random variables, with variance $\boldsymbol{Q}$ and $\boldsymbol{R}$ respectively [20] [21] [22] [23] [24]. An arbitrary linear dynamic system, and corresponding measurement mechanism, that satisfies these equations can be modeled as shown in Figure 2-4.

$$\boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1} + \boldsymbol{w}_{k-1} \tag{2-8}$$

$$\tilde{\boldsymbol{z}}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{v}_k \tag{2-9}$$

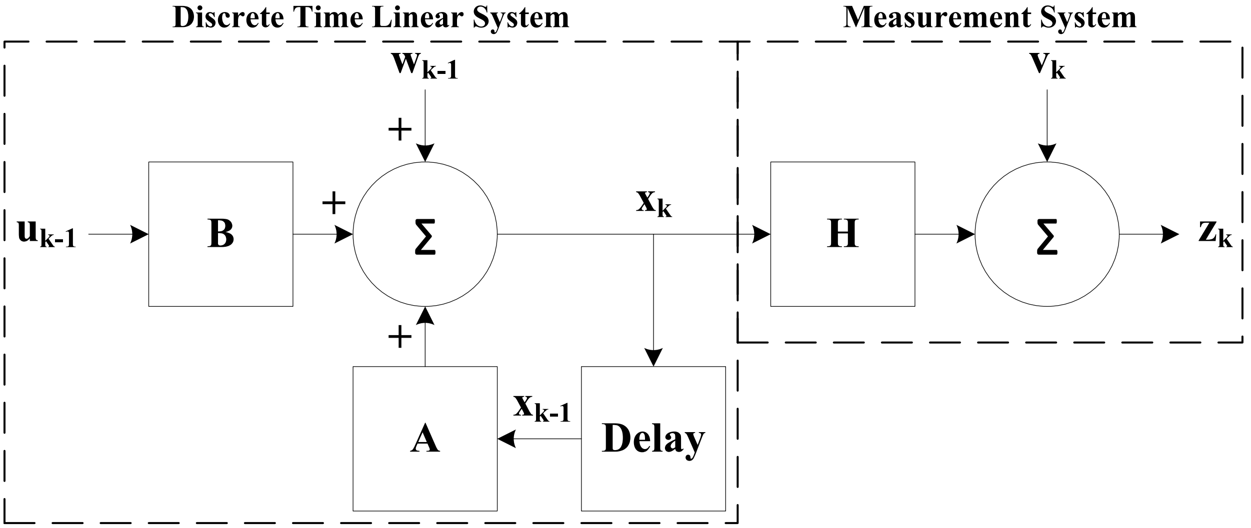

Figure 2-4: Discrete Time Linear System and Measurement Device [22].

A block diagram of the Kalman filter process is provided in Figure 2-5. Note that the Kalman filter includes a model of the linear system being estimated (shown in the lower right of Figure 2-5) and the input to the system ($\boldsymbol{z}_k$) is the discrete time measured state of the physical linear system shown in Figure 2-4. The Kalman filter utilizes feedback to estimate the state of the system by generating a prediction of the system state based on the Kalman filter's system model. Feedback is then obtained from the measured state of the physical linear system to correct the prediction [20] [21] [22] [23] [24]. Accordingly, the Kalman filter equations fall into two categories: prediction (sometimes called time propagation) equations and correction (also known as observation update) equations [20] [21] [22] [23] [24]. The prediction equations provide an *a* *priori* estimate of the system, while the correction equations incorporate new measurements into the *a priori* estimate, resulting in an improved *a posteriori* estimate [20].

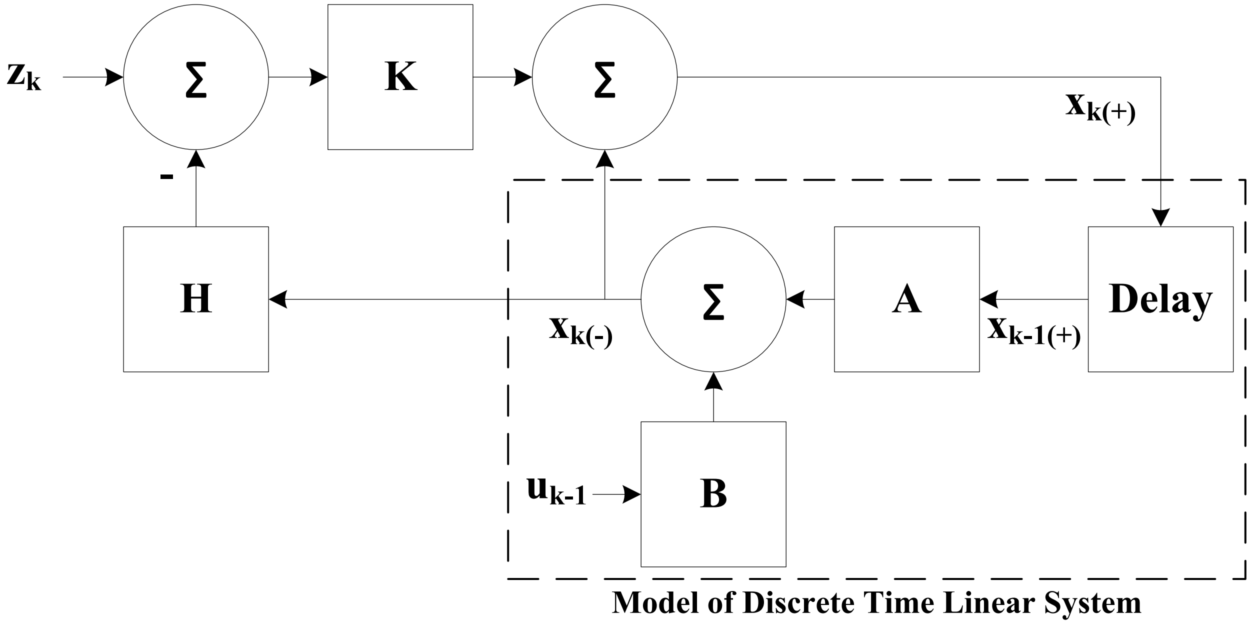

Figure 2-5: Discrete Kalman Filter Model [22].

The state and covariance estimation equations governing the prediction process are given in (2-10) and (2-11); while the state and covariance equations related to the correction process are shown in (2-12), (2-13), and (2-14) [20] [21] [22] [23] [24]. A complete derivation of these equations is available in [21] and [23].

Kalman Filter State and Covariance Prediction Equations

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{A}\hat{\boldsymbol{x}}_{k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1} \tag{2-10}$$

$$\boldsymbol{P}_k^- = \boldsymbol{A}\boldsymbol{P}_{k-1}\boldsymbol{A}^T + \boldsymbol{Q} \tag{2-11}$$

Kalman Filter State and Covariance Correction Equations

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\tilde{\boldsymbol{z}}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_k^-) \tag{2-12}$$

$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H})\boldsymbol{P}_k^- \tag{2-13}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_k^-\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{P}_k^-\boldsymbol{H}^T + \boldsymbol{R})^{-1} \tag{2-14}$$

Prior to receiving a measurement, and during times of measurement unavailability, the prediction calculations – equations (2-10) and (2-11) – are performed to generate an estimate of the *a priori* state and covariance of the system. Upon receipt of a valid measurement, the correction calculations – equations (2-12), (2-13), and (2-14) – are performed. These equations determine the difference between the predicted state and the measured state, form a correction component in the form of Kalman Gain, $\boldsymbol{K}$, and update the state and covariance to provide an improved *a posteriori* estimate of the system.

2.3.1 Extended Kalman Filter

The first known application of the Kalman filter was trajectory estimation and control of the Apollo space project, using what would today be known as an Extended Kalman filter (EKF) [22]. The EKF seeks to estimate the state and covariance of a non-linear discrete-time process by linearizing the system about the current state estimate [20]. This extension of the linear Kalman filter to non-linear applications was recognized by Stanley F. Schmidt while working at the NASA Ames Research Center [22].

The symbolic notation previously provided in Table 2-12 remains relevant to the EKF, with a few minor changes and additions that will be discussed as introduced throughout the remainder of this section.

The EKF seeks to estimate the state and covariance of a non-linear discrete time process defined by the stochastic difference equation in (2-15), given a measurement of the form expressed in (2-16) [20] [21] [22] [23] [24]. Note that $\boldsymbol{x}_k$and $\tilde{\boldsymbol{z}}_k$ are now expressed as functions $\boldsymbol{f}$ and $\boldsymbol{h}$ respectively. They can no longer be expressed as a system of linear equations because the non-linear variables and coefficients are no longer separable. It should be noted that $\boldsymbol{f}$ and $\boldsymbol{h}$ are vectors, where each element defines the non-linear equation required to calculate the corresponding state/measurement value. This is expressed in the expanded equation forms of (2-15) and (2-16).

$$\boldsymbol{x}_k = \boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}, \boldsymbol{w}_{k-1}) \Rightarrow \begin{bmatrix} x_{1_k} \\ x_{2_k} \\ \cdot \\ \cdot \\ \cdot \\ x_{n_k} \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}, \boldsymbol{w}_{k-1}) \\ f_2(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}, \boldsymbol{w}_{k-1}) \\ \cdot \\ \cdot \\ \cdot \\ f_n(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}, \boldsymbol{w}_{k-1}) \end{bmatrix} \qquad \textbf{(2-15)}$$

$$\tilde{\boldsymbol{z}}_k = \boldsymbol{h}(\boldsymbol{x}_{k,}\boldsymbol{v}_k) \Rightarrow \begin{bmatrix} z_{1_k} \\ z_{2_k} \\ \cdot \\ \cdot \\ \cdot \\ z_{n_k} \end{bmatrix} = \begin{bmatrix} h_1(\boldsymbol{x}_k, \boldsymbol{v}_k) \\ h_2(\boldsymbol{x}_k, \boldsymbol{v}_k) \\ \cdot \\ \cdot \\ \cdot \\ h_n(\boldsymbol{x}_k, \boldsymbol{v}_k) \end{bmatrix} \qquad \textbf{(2-16)}$$

The EKF retains the model and computational process of a Kalman filter by generating an *a priori* estimate, then refining the estimate as measurement data becomes available to generate an *a posteriori* estimate [20]. However, the non-linear nature of the EKF system model precludes the use of a system of linear equations expressed in matrix form. To overcome this, the estimate is linearized through the calculation of Jacobians. A Jacobian is the partial derivative of the non-linear function with respect to each state element [20] [21] [22] [23] [24]. This leads to a redefinition of the matrices $\boldsymbol{A}$ and $\boldsymbol{H}$ as shown in (2-17) and (2-18). The expanded notation of equation (2-17) is presented to clarify the double indexed *i, j* notation used here and in other Jacobian equations expressed in this section. For completeness, the matrices $\boldsymbol{W}$ and $\boldsymbol{V}$ are introduced in (2-19) and (2-20) respectively [20]. These matrices can be used to linearize the process and measurement uncertainty.

$$\boldsymbol{A}_{[i,j]} = \frac{\partial \boldsymbol{f}_{[i]}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}, \boldsymbol{w}_{k-1})}{\partial \boldsymbol{x}_{[j]}} = \begin{bmatrix} \frac{\partial f_{[1]}}{\partial x_{[1]}} & \frac{\partial f_{[1]}}{\partial x_{[2]}} & \cdot\ \cdot\ \cdot & \frac{\partial f_{[1]}}{\partial x_{[n]}} \\ \frac{\partial f_{[2]}}{\partial x_{[1]}} & \frac{\partial f_{[2]}}{\partial x_{[2]}} & \cdot\ \cdot\ \cdot & \frac{\partial f_{[2]}}{\partial x_{[n]}} \\ \cdot & \cdot & \cdot\ \cdot\ \cdot\ \cdot & \cdot \\ \cdot & \cdot & \cdot\ \cdot\ \cdot\ \cdot & \cdot \\ \cdot & \cdot & \cdot\ \cdot\ \cdot\ \cdot & \cdot \\ \frac{\partial f_{[n]}}{\partial x_{[1]}} & \frac{\partial f_{[n]}}{\partial x_{[2]}} & \cdot\ \cdot\ \cdot & \frac{\partial f_{[n]}}{\partial x_{[n]}} \end{bmatrix} \tag{2-17}$$

$$\boldsymbol{H}_{[i,j]} = \frac{\partial \boldsymbol{h}_{[i]}(\boldsymbol{x}_k, \boldsymbol{v}_k)}{\partial \boldsymbol{x}_{[j]}} \tag{2-18}$$

$$\boldsymbol{W}_{[i,j]} = \frac{\partial \boldsymbol{f}_{[i]}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}, \boldsymbol{w}_{k-1})}{\partial \boldsymbol{w}_{[j]}} \tag{2-19}$$

$$\boldsymbol{V}_{[i,j]} = \frac{\partial \boldsymbol{h}_{[i]}(\boldsymbol{x}_k, \boldsymbol{v}_k)}{\partial \boldsymbol{v}_{[j]}} \tag{2-20}$$

Applying the various Jacobian matrices to the Kalman filter equations results in the equations governing the prediction process being redefined as shown in (2-21) and (2-22); while the equations related to the correction process are redefined in (2-23), (2-24), and (2-25) [20] [21] [22] [23] [24]. A more complete derivation of these equations is available in [21] and [23].

EKF State and Covariance Prediction Equations

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_{k-1}, 0) \tag{2-21}$$

$$\boldsymbol{P}_k^- = \boldsymbol{A}_k \boldsymbol{P}_{k-1} \boldsymbol{A}_k^T + \boldsymbol{W}_k \boldsymbol{Q}_{[k-1]} \boldsymbol{W}_k^T \tag{2-22}$$

EKF State and Covariance Correction Equations

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k[\tilde{\boldsymbol{z}}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_k^-, 0)] \tag{2-23}$$

$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_k^- \tag{2-24}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}_k^T \left(\boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^T + \boldsymbol{V}_k \boldsymbol{R}_k \boldsymbol{V}_k^T \right)^{-1} \tag{2-25}$$

It is worth noting that the EKF and linear Kalman filter are not mutually exclusive in their implementation. It is not necessary to linearize the aspects of the system that are already linear just to fit the system into an EKF model. The Kalman filter and EKF can be combined/tailored to the specific application such that the EKF equations are applied to the non-linear portions of the system, while the standard linear Kalman equations are applied to the linear portions [24]. This makes the implementation remarkably flexible.

2.3.2 ***Schmidt-Kalman Filter***

In addition to his contributions regarding the EKF, Schmidt made a number of other significant contributions to the field of Kalman filtering. One such contribution, known as the Schmidt-Kalman filter, seeks to reduce computational complexity of the filter by eliminating nuisance variables from state estimation while still taking their uncertainties into account [22] [23]. In this context, nuisance variables are those state variables that are of no interest to the problem, but must remain in the state vector [22]. Examples of these nuisance variables might include measurement biases, non-Gaussian measurement noise, or other parameters that are not dynamically coupled to other state variables [22]. That is, the state transition matrix does not link these nuisance variables to the states of interest. This method is ideal for accounting for the effects of unobservable biases.

In a Schmidt-Kalman filter design, the nuisance variables are segregated from the essential variables and placed last in the state vector as demonstrated in (2-26) [22].

$$\boldsymbol{x}_a = [\boldsymbol{x}_e \quad \boldsymbol{x}_n]^T = [x_{e_1} \quad x_{e_2} \quad \cdots \quad x_{e_i} \quad x_{n_1} \quad x_{n_2} \quad \cdots \quad x_{n_j}]^T \tag{2-26}$$

Where: $\boldsymbol{x}_a$ = Augmented state vector.

$\boldsymbol{x}_e$ = Essential system states.

$\boldsymbol{x}_n$ = Nuisance system states.

This segregation is only possible because the nuisance variables are not dynamically coupled to the essential states in the system model. This is reflected in (2-27) by the fact that the off-diagonal terms are zero [22]. For clarity, it should be noted that (2-27) is a restatement of (2-8) with no control input. This partitioning of essential variables from nuisance variables is then applied across all Kalman filter matrices.

$$\boldsymbol{x}_{a_k} = \boldsymbol{A}_a \boldsymbol{x}_{a_{k-1}} + \boldsymbol{w}_{k-1} = \begin{bmatrix} \boldsymbol{A}_e & 0 \\ 0 & \boldsymbol{A}_n \end{bmatrix} \boldsymbol{x}_{a_{k-1}} + \boldsymbol{w}_{k-1} \tag{2-27}$$

Where: $\boldsymbol{A}_a$ = Augment state transition matrix.
$\boldsymbol{A}_e$ = Essential state transition dynamics.
$\boldsymbol{A}_n$ = Uncoupled nuisance state dynamics.

Because the desire is to reduce computational complexity, the nuisance variable states are not estimated, but their influence is considered in the overall filter structure through their effect on the covariance matrix via cross correlations between the filter states [22]. To achieve this, a sub-optimal Kalman gain is utilized in the system state measurement update equations. This sub-optimal gain has the corresponding nuisance variable gains set to 0 [22]; which has the effect of holding the nuisance variables states at 0. The optimal Kalman gain is still used in the covariance update equations so that the nuisance variables affect the overall system uncertainty without directly estimating their state. The optimal partitioned Kalman gain used in the covariance update equations is given as (2-28) [22]; while the sub-optimal partitioned Kalman gain used in the state update equations in given as (2-29) [22]. Likewise, the sub-optimal state update equation is presented as (2-30) [22]; while a modified covariance update equation is shown in (2-31) [22]. Note that (2-30) is simply a restatement of (2-12) using the sub-optimal Kalman gain. Grewal and Andrews provide a more complete derivation of the Schmidt-Kalman filter in [22].

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_e \\ \boldsymbol{K}_n \end{bmatrix} \quad \textbf{(2-28)}$$

Where: $\boldsymbol{K}$ = Optimal Kalman gain, computed using (2-14) or (2-25).
$\boldsymbol{K}_e$ = Essential state Kalman gain.
$\boldsymbol{K}_n$ = Uncoupled nuisance state Kalman gain.

$$\boldsymbol{K}_s = \begin{bmatrix} \boldsymbol{K}_e \\ 0 \end{bmatrix} \quad \textbf{(2-29)}$$

Where: $\boldsymbol{K}_s$ = Sub-optimal Kalman gain.
$\boldsymbol{K}_e$ = Essential state Kalman gain.
$\boldsymbol{K}_n$ = 0.

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_{s_k}(\tilde{\boldsymbol{z}}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_k^-) \quad \textbf{(2-30)}$$

$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_k^-(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)^T + \boldsymbol{K}_k\boldsymbol{R}_k\boldsymbol{K}_k^T \quad \textbf{(2-31)}$$

2.4 Simultaneous Localization and Mapping

SLAM is an algorithmic process that allows a mobile vehicle to generate a map of the environment while simultaneously using the map to determine the location of the vehicle within the mapped environment [25]. In a SLAM implementation, both the motion of the vehicle and the location of the landmarks are estimated without previous knowledge of their locations [25]. An introductory tutorial of SLAM is available in [25], a summary of which is provided here. Preliminary notion relative to SLAM is provided in Table 2-13.

Table 2-13: Notation Applicable to SLAM [25].

Symbol	Description
$\boldsymbol{u}_k$	Control input at time k
$\boldsymbol{v}_k$	Additive zero mean white noise with covariance $\boldsymbol{R}$
$\boldsymbol{w}_k$	Additive zero mean white noise with covariance $\boldsymbol{Q}$
$\boldsymbol{x}_0$	Initial vehicle state vector at time 0
$\boldsymbol{x}_k$	Vehicle state vector at time k
$\boldsymbol{z}_k$	Observation at time k
$\mathbb{M}$	$\{m_1, m_2, \dots, m_n\}$ Complete set of all landmarks
$\mathbb{U}$	$\{u_1, u_2, \dots, u_k\}$ Complete history of control inputs
$\mathbb{Z}$	$\{z_1, z_2, \dots, z_k\}$ Complete set of all observations
$\boldsymbol{f}(\cdot)$	Vehicle motion model function
$\boldsymbol{h}(\cdot)$	Measurement geometry function
$\boldsymbol{A}$	State transition matrix (Jacobian of $\boldsymbol{f}$)
$\boldsymbol{H}$	Measurement matrix (Jacobian of $\boldsymbol{h}$)

Expressed in probabilistic terms, SLAM attempts to determine the probability density function (PDF) given in (2-32) [25]. This density function describes the joint distribution of the vehicle state and landmark locations given the initial vehicle state, all control inputs, and all observations [25].

$$P(\boldsymbol{x}_k, \mathbb{M} | \mathbb{Z}_k, \mathbb{U}_k, \boldsymbol{x}_0) \tag{2-32}$$

Given the complexity of computing (2-32) for all times *k*, a recursive solution is essential for real-time applications [25]. Assuming the joint probability distribution is known for time *k* - 1, a control input $\boldsymbol{u}_k$ is provided, and an observation $\tilde{\boldsymbol{z}}_k$ is available, then the joint distribution can be determined using Bayes theorem [25]. Prior to applying Bayes theorem, a state transition model and an observation model must be defined.

The state transition model describes the motion of the vehicle, and it is assumed to be a Markov process [25]. This allows the next state to be determined using only the previous state and the control input $\boldsymbol{u}_k$, independent of both the map and observations [25]. This is expressed in probabilistic terms in (2-33).

$$P(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}, \boldsymbol{u}_k) \tag{2-33}$$

The observation model, given in (2-34), expresses the probability of the observation, z_k, when the system state and landmark locations are known [25].

$$P(\tilde{\boldsymbol{z}}_k | \boldsymbol{x}_k, \mathbb{M}) \tag{2-34}$$

The SLAM algorithm can now be expressed as a recursive prediction/correction process, shown in (2-35) and (2-36), where (2-35) describes the prediction and (2-36) describes the correction step [25]. Together, these equations determine the joint distribution for the vehicle state and landmark locations based on the set of all observations and control inputs [25].

$$P(\boldsymbol{x}_k, \mathbb{M} | \mathbb{Z}_{k-1}, \mathbb{U}_k, x_0) = \int P(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}, \boldsymbol{u}_k) P(\boldsymbol{x}_{k-1}, \mathbb{M} | \mathbb{Z}_{k-1}, \mathbb{U}_{k-1}, x_0) d\boldsymbol{x}_{k-1} \tag{2-35}$$

$$P(\boldsymbol{x}_k, \mathbb{M}|\mathbb{Z}_{k-1}, \mathbb{U}_k, x_0) = \frac{P(\boldsymbol{z}_k|\boldsymbol{x}_k, \mathbb{M})P(\boldsymbol{x}_k, \mathbb{M}|\mathbb{Z}_{k-1}, \mathbb{U}_k, x_0)}{P(\boldsymbol{z}_k|\mathbb{Z}_{k-1}, \mathbb{U}_k)} \tag{2-36}$$

It is common to express the SLAM problem in terms of a state-space model with additive Gaussian noise, then to solve the problem using an EKF [25]. In EKF based solutions, the motion model of the vehicle, previously shown in (2-33), can be expressed as given in (2-37); and the observation model, shown in (2-34), can be expressed as given in (2-38) [25].

$$P(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{u}_k) \Leftrightarrow \boldsymbol{x}_k = \boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k) + \boldsymbol{w}_k \tag{2-37}$$

$$P(\boldsymbol{z}_k|\boldsymbol{x}_k, \mathbb{M}) \Leftrightarrow \boldsymbol{z}_k = \boldsymbol{h}(\boldsymbol{x}_{k,}\mathbb{M}) + \boldsymbol{v}_k \tag{2-38}$$

The resulting EKF time update equations then take on the familiar form presented in (2-39) and (2-40). Similarly, the EKF observation update equations are as given in (2-41) through (2-43), noting that (2-41) has been extended from the form given in (2-23) to simultaneously consider the vehicle state and landmark locations.

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_k) \tag{2-39}$$

$$\boldsymbol{P}_k^- = \boldsymbol{A}_k \boldsymbol{P}_{k-1} \boldsymbol{A}_k^T + \boldsymbol{Q}_k \tag{2-40}$$

$$\begin{bmatrix} \hat{\boldsymbol{x}}_k \\ \widehat{\mathbb{M}}_k \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{x}}_k^{-T} & \widehat{\mathbb{M}}_{k-1}{}^T \end{bmatrix}^T + \boldsymbol{K}_k \left[\tilde{\boldsymbol{z}}_k - \boldsymbol{h}\left(\hat{\boldsymbol{x}}_k^-, \widehat{\mathbb{M}}_{k-1}\right) \right] \tag{2-41}$$

$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_k^- \tag{2-42}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}_k^T \left(\boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^T + \boldsymbol{R}_k \right)^{-1} \tag{2-43}$$

Care was taken with Equation (2-41) to explicitly separate the estimated vehicle state $\hat{\boldsymbol{x}}_k$ from the estimate of the landmark locations $\widehat{\mathbb{M}}_k$. This was done to show that both estimates are determined simultaneously. An alternate expression for the system state could be written that combines the vehicle state and landmark locations into a single system state as shown in (2-44).

$$\hat{\boldsymbol{x}}_k = \left[\hat{\boldsymbol{x}}_{v_k}{}^T \quad \widehat{\mathbb{M}}_k{}^T\right]^T \qquad \textbf{(2-44)}$$

Where: $\hat{\boldsymbol{x}}_k$ = Combined vehicle and landmark state estimate at time k.

$\hat{\boldsymbol{x}}_{v_k}$ = Vehicle state estimate at time k.

$\widehat{\mathbb{M}}_k$ = Landmark estimates at time k.

Using the combined notation of (2-44), the EKF equations would remain unchanged from the form presented in Section 2.3.1; however, this may obfuscate the fact that the system is simultaneously updating the estimate of both the vehicle and landmark locations.

In spite of the abundant research conducted on the SLAM algorithm, it is still subject to numerous challenges and ongoing research. Three SLAM related challenges are: the size of the EKF covariance matrix, data association, and optimistic uncertainty estimates [26] [27]. These challenges are summarized in the remaining paragraphs of this section.

As the vehicle moves through its environment, it encounters new landmarks. Each new landmark necessitates an expansion of the system state by two elements – landmark position x and y – to facilitate tracking and enhancement of the landmark's estimated position. The addition of two new states to the system results in the covariance matrix growing quadratically [26]. This large, ever increasing covariance matrix is typically non-sparse, as the off-diagonal elements represent the correlations between different random variables defined in the system [26]. The non-sparseness of the matrix limits the computational techniques that can be used to increase the efficiency of the calculations involving this matrix.

With respect to data association, the EKF-SLAM algorithm is particularly sensitive to incorrect association of observations to a landmark [25]. That is, applying an observation to the wrong landmark can seriously affect the accuracy of the system. Typically, this is overcome by proximity evaluation [26]. That is, the algorithm determines which landmark is the most likely landmark observed based on its proximity to other known landmarks [26]. Another common technique to ensure correct data association is to withhold a landmark from the map or to withhold the measurement until the landmark has been observed a sufficient number of times to determine if the observation if of a new landmark or an existing one [26].

Huang and Dissanayake noted that a number of researchers have observed inconsistency in EKF SLAM implementations where the SLAM algorithm produces optimistic (over confident) estimates [27]. Many researchers believe that linearization is the root cause of the optimistic estimates, but this has yet to be proven for large scale maps, and continues to be an area of ongoing research [27].

2.5 Random Variables and Statistical Conversions

Often when dealing with random variables it becomes necessary to convert from one probability level to another within a given distribution, or to express a random variable given in one distribution as a random variable in a different distribution. For example, GNSS uncertainty is often expressed as a 99.999% radius of containment, but some operations that perform calculations on this data require the uncertainty to be expressed as a Gaussian distributed random variable. This section derives the process to convert from one Gaussian probability level to another, to convert from Gaussian to Rayleigh, and to convert between Rayleigh probability levels. For clarity, the notation σ^{DP} will be used when expressing the standard deviation of a random variable. The superscript 'D' will denote the distribution as either 'N' for Normal (Gaussian) or 'R' for Rayleigh; while the superscript 'P' will indicate the probability level. For example, σ^{R95} indicates the standard deviation for a Rayleigh random variable at the 95% probability level.

2.5.1 Gaussian Random Variables

A Gaussian random variable, often referred to as a Normal random variable, has a Probability Density Function (PDF) that forms the well-known "bell-curve" [28]. The Gaussian random variable is often used to characterize physical quantities, and is particularly useful when a large sample size is available [28]. The Gaussian distribution is defined by two parameters: the mean, μ, and the standard deviation, σ; where $-\infty < \mu < \infty$ and $\sigma > 0$ [29]. The notation $N \sim (\mu, \sigma)$ is often used to describe a Gaussian distribution. The PDF of a Gaussian random variable is given in (2-45) [28] [29].

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2-45}$$

Where: μ = Mean

σ = Standard deviation

σ^2 = Variance

A plot of the PDF of the Gaussian distribution $N \sim (0, 1)$ is provided in Figure 2-6. The vertical cyan lines denote ± one standard deviation from the mean. As detailed later in this section, approximately 68% of all outcomes will fall inside of this band. It is clear from this plot that the mean is the most likely outcome – or expected value – of the distribution. It is also evident that the mean represents the midpoint of the distribution, with 50% of all possible outcomes falling below the mean, and 50% of all possible outcomes falling above the mean. Clearly, the expected value, median, and mode of a Gaussian distribution are all equal to the mean μ.

The standard deviation provides an indication of the level of dispersion of the data. Large values of σ result in a shorter and wider bell-curve, while smaller values of σ lead to a tall and narrow curve [29]. Figure 2-7 shows the effect of the standard deviation on the shape of the distribution.

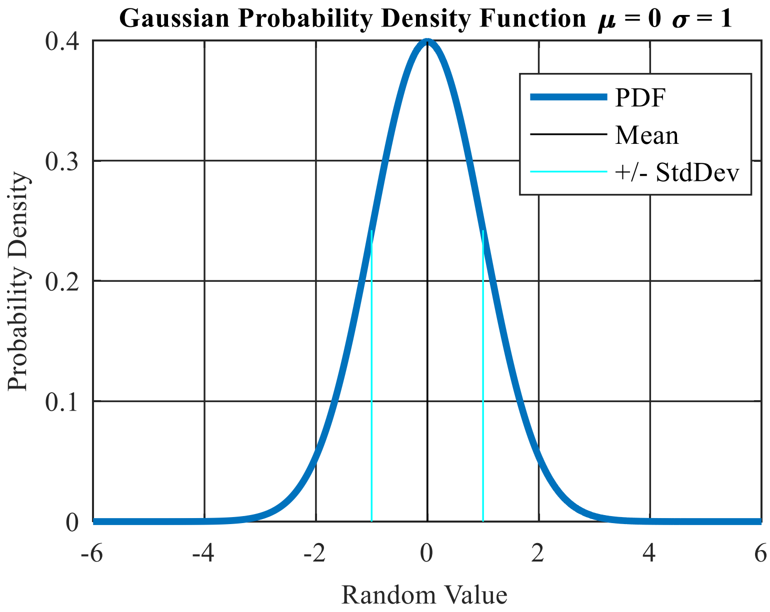

Figure 2-6: Gaussian Probability Density Function with $\mu = 0$ and $\sigma = 1$.

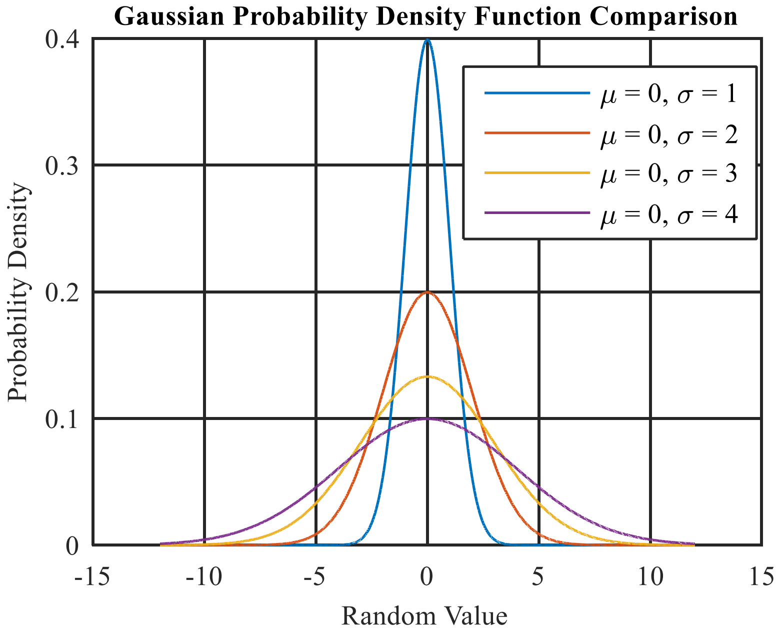

Figure 2-7: Comparison of Gaussian Distributions with Varying Standard Deviations.

The CDF of a Gaussian random variable, expressed in (2-46), does not have a closed form solution; therefore, the error function expressed in (2-47) is often used to simplify equations involving the CDF of a Normally distributed random variable [28].

$$F_X(x) = \int_{-\infty}^{x} f_X(v)dv \;=\; \frac{1}{\sqrt{2\pi\sigma^2}}\int_{-\infty}^{x} e^{-\frac{(v-\mu)^2}{2\sigma^2}}dv \tag{2-46}$$

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_{0}^{x} e^{(-v^2)}dv \tag{2-47}$$

Substituting (2-47) into (2-46) results in an expression of the CDF of a Normal random variable in terms of the error function; this result is provided in (2-48).

$$F_X(x) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right] \tag{2-48}$$

The error function is an odd function; accordingly, it exhibits the property in (2-49) [28].

$$\mathrm{erf}(-x) = -\mathrm{erf}(x) \tag{2-49}$$

Amongst other things, the CDF can be used to determine the percent probability that a random value *X*, drawn from a Gaussian distribution, falls within n standard deviations of the mean. That is, the CDF can be used to determine the probability that *X* falls in the range $[(\mu - n\sigma),(\mu + n\sigma)]$. This assertion is derived in (2-50) and (2-51). Note that the Normal distribution is a two-sided function, implying that the contribution from both sides of the y-axis must be considered as shown in (2-50).

$$\begin{aligned} P(n) &= \; F(\mu + n\sigma) - F(\mu - n\sigma) \\ &= \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{\mu + n\sigma - \mu}{\sigma\sqrt{2}}\right)\right] - \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{\mu - n\sigma - \mu}{\sigma\sqrt{2}}\right)\right] \\ &= \frac{1}{2}\left[\mathrm{erf}\left(\frac{n}{\sqrt{2}}\right) - \mathrm{erf}\left(-\frac{n}{\sqrt{2}}\right)\right] \end{aligned} \tag{2-50}$$

Applying (2-49) to (2-50) results in an expression for the probably level as a function of n. This is shown in (2-51).

$$P(n) = \; \mathrm{erf}\left(\frac{n}{\sqrt{2}}\right) \tag{2-51}$$

The probability level for several common sigma multiples computed using (2-51) are given in Table 2-14. This table shows that the probability of a random sample being within 1σ of the mean is approximately 68%, of being within 2σ of the mean is approximately 95.5%, and of being within 3σ of the mean is approximately 99.7%.

Table 2-14: Gaussian Probability Levels for $n\sigma$.

n	P [%]	n	P [%]
1	68.268949	4	99.993666
2	95.449974	5	99.999943
3	99.730020	6	99.999999

Similarly, the inverse error function can be used to determine the sigma multiple n required to reflect a given probability level. The inverse error function is defined to have the property shown in (2-52). Applying this property to (2-51) results in (2-53).

$$\mathrm{erf}^{-1}[\mathrm{erf}(x)] = x \tag{2-52}$$

$$n = \sqrt{2}[\mathrm{erf}^{-1}(P)] \tag{2-53}$$

The sigma multiple n for some common probability levels was determined using (2-53) and are given in Table 2-15.

Table 2-15: Sigma Multiple for Gaussian Probability Levels.

P [%]	n	P [%]	n
68.268949	1	97.500000	2.241403
90.000000	1.644854	99.000000	2.575829
92.500000	1.780464	99.500000	2.807034
95.000000	1.959964	99.900000	3.290527

Table 2-15 indicates that conversion from a 1σ Gaussian random variable to any desired probability level can be accomplished through multiplication by the appropriate factor of n. Equation (2-54) demonstrates the conversion to express a 1σ Gaussian random variable as a Gaussian random variable enclosing 95% of all samples.

$$\sigma^{N95} = 1.959964\sigma^{N68} \tag{2-54}$$

2.5.2 Rayleigh Random Variables

The Rayleigh random variable is common in physical phenomena and various applications where random variables are summed, such as: noise theory, processing radar returns, wireless communication, ultrasound, and wind modelling [28] [30]. The Rayleigh distribution is completely defined by a single parameter α, known as the scale factor [28] [30]. The general form of the Rayleigh PDF and CDF are given in (2-55) and (2-56) respectively, where $\alpha > 0$ [28].

$$f_X(x) = \left(\frac{x}{\alpha^2}\right) e^{-\frac{x^2}{2\alpha^2}} \tag{2-55}$$

$$F_X(x) = \int_0^x f_X(v)dv = 1 - e^{-\frac{x^2}{2\alpha^2}} \tag{2-56}$$

Figure 2-8 provides a plot of the Rayleigh PDF with $\alpha = 1$. It is clear from this plot that the mode of the distribution is α, and that the one sided nature of the distribution causes the mean and median to differ from the mode. (2-57) and (2-58) define the relationship between the mode, mean, and median of the Rayleigh distribution, while (2-59) defines the relationship between the mode and variance of a Rayleigh distribution [30].

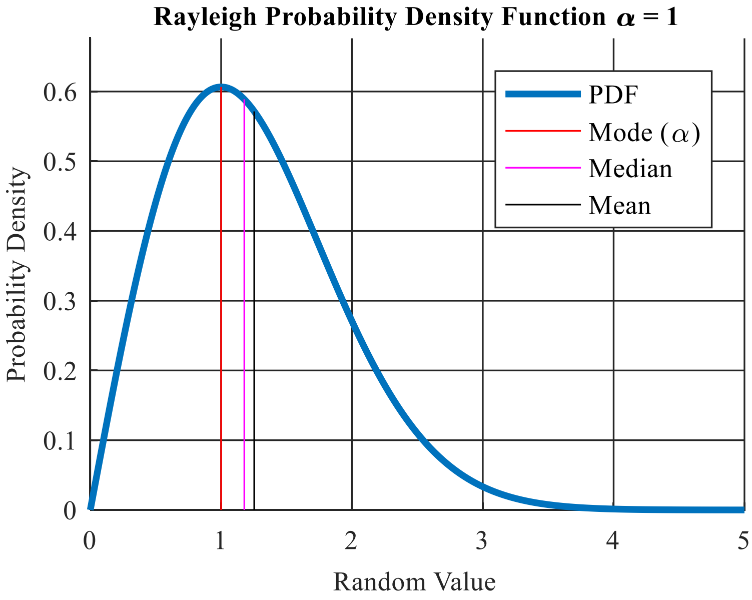

Figure 2-8: Rayleigh Probability Density Function with $\alpha = 1$.

$$\bar{x} = \alpha\sqrt{\frac{\pi}{2}} \qquad \textbf{(2-57)}$$

Where: $\bar{x}$ = Mean of a Rayleigh distribution.

α = Scale factor = Mode of a Rayleigh distribution.

$$\tilde{x} = \alpha\sqrt{2\ln(2)} \qquad \textbf{(2-58)}$$

Where: $\tilde{x}$ = Median of a Rayleigh distribution.

$$var_R = \frac{4 - pi}{2}\alpha^2 \qquad \textbf{(2-59)}$$

Where: var_R = Variance of a Rayleigh distribution.

Similar to σ in the Gaussian case, the scale factor α affects the height and width of a Rayleigh distribution. The Rayleigh PDF for several values of α are shown in Figure 2-9.

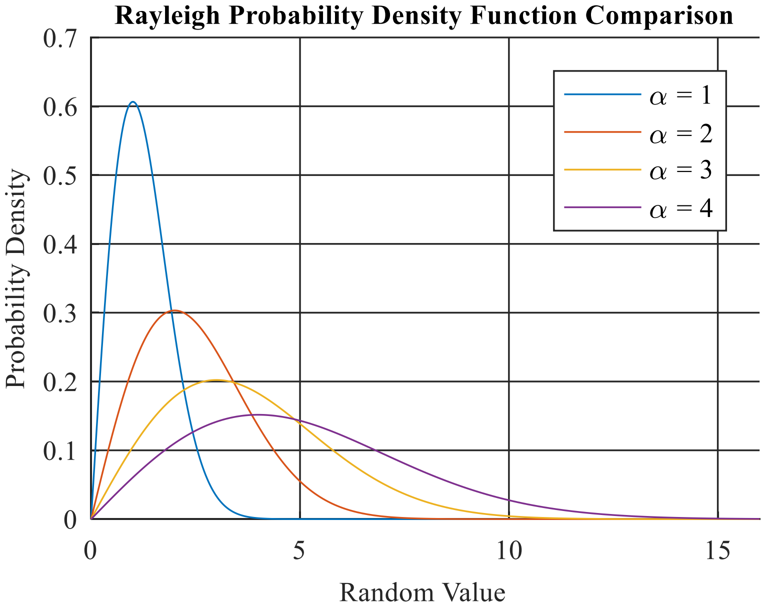

Figure 2-9: Comparison of Rayleigh Distributions with Varying Scale Factors.

If the Rayleigh distribution is formed as the sum of zero mean, independent Gaussian random variables X and Y, with standard deviations σ_x and σ_y, such that $\sigma_x = \sigma_y = \sigma$, then $\alpha = \sigma$ and $Z = \sqrt{X^2 + Y^2}$ is a Rayleigh random variable with the PDF and CDF shown in (2-60) and (2-61) respectively [28].

$$f_X(x) = \left(\frac{x}{\sigma^2}\right) e^{-\frac{x^2}{2\sigma^2}} \tag{2-60}$$

$$F_X(x) = \int_0^x f_X(v)dv = 1 - e^{-\frac{x^2}{2\sigma^2}} \tag{2-61}$$

Inherent in this method of constructing a Rayleigh distribution is that fact that the mode α, is equal to the standard deviation σ, of the underlying Gaussian distribution. This allows for convenient conversion between a Rayleigh distribution and it constituent Gaussian distribution. The remainder of this section (and this document) assumes that Rayleigh random variables are the sum of zero mean Gaussian random variables. On this note, a point of clarification is in order regarding the notion for the standard deviation of a Rayleigh random variable introduced in Section 2.5. The notation σ^R refers to the standard deviation of the underlying Gaussian distribution used to construct the Rayleigh distribution, it does not refer to the square root of the variance expressed in (2-59). This liberty with the notation allows convenient conversion between a Rayleigh distribution and its constituent Gaussian distribution.

To demonstrate the relationship between the Gaussian distribution and the corresponding Rayleigh distribution, Figure 2-10 provides a scatter plot of 10,000 Rayleigh distributed random (X,Y) pairs that were constructed as the sum of Gaussian random variables. In this sample, the standard deviation of the underlying Gaussian distribution was 1. This plot shows the radial nature of the distribution, with the red circle denoting the underlying Gaussian 1σ containment level. The Gaussian 1σ containment level encloses approximately 39% of possible Rayleigh outcomes.

As in the Gaussian case, the Rayleigh CDF can be used to determine the exact percent probability that a random value X, drawn from a Rayleigh distribution, will fall within n standard deviations of the mean. In this case, the standard deviation refers to the underlying

Gaussian standard deviation. Unlike the Gaussian case, the Rayleigh distribution is one-sided, as evident in Figure 2-8. Equation (2-62) derives the equation for this process, and Table 2-16 shows the probability level for several values of $n\sigma^{N68}$ as computed using (2-62).

$$P(n) = F(n\sigma^{N68}) = 1 - e^{-\frac{n^2\sigma^2}{2\sigma^2}} = 1 - e^{-\frac{n^2}{2}} \tag{2-62}$$

Table 2-16: Rayleigh Probability Levels for *nσ*.

$n\sigma^{N68}$	P$_R$ [%]	$n\sigma^{N68}$	P$_R$ [%]
$1\sigma^{N68}$	39.346934	$4\sigma^{N68}$	99.966454
$2\sigma^{N68}$	86.466472	$5\sigma^{N68}$	99.999627
$3\sigma^{N68}$	98.889100	$6\sigma^{N68}$	99.999998

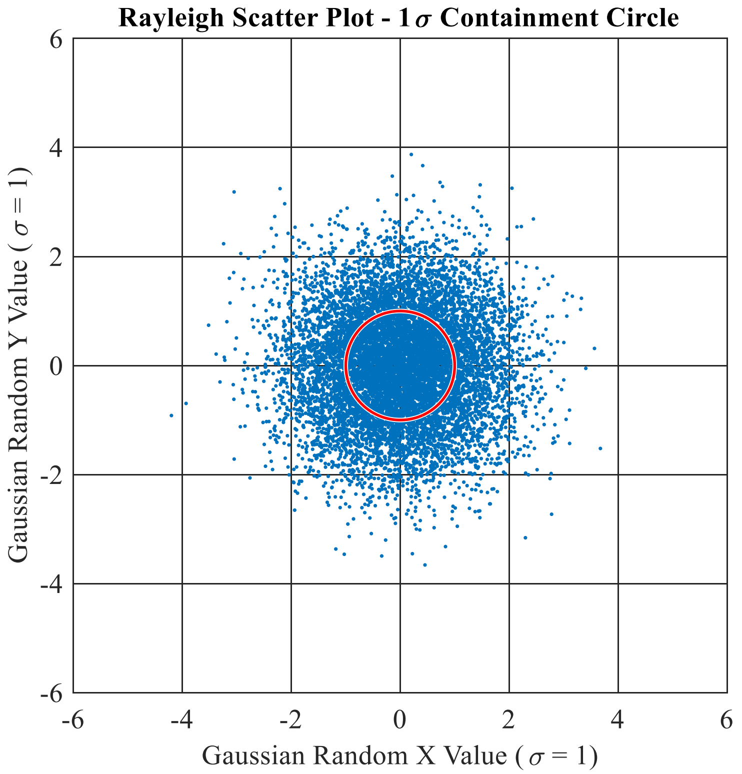

Figure 2-10: Rayleigh Random Variable Scatter Plot with σ = 1.

As in the Normal distribution case, the inverse CDF, given in (2-63), can be used to determine the Gaussian sigma multiple required to enclose a given Rayleigh probability level. The sigma multiple, *n*, for some common Rayleigh probability levels was determined using (2-63) and are given in Table 2-17.

$$F_X(n\sigma^{N68}) = 1 - e^{-\frac{n^2}{2}} => e^{-\frac{n^2}{2}} = 1 - F_X => \ n = \sqrt{-2\ln(1 - F_X)} \qquad \textbf{(2-63)}$$

Table 2-17: Sigma Multiple for Rayleigh Probability Levels.

P_R [%]	$n\sigma^{N68}$	P_R [%]	$n\sigma^{N68}$
68.268949	$1.515173\sigma^{N68}$	97.500000	$2.716203\sigma^{N68}$
90.000000	$2.145966\sigma^{N68}$	99.000000	$3.034854\sigma^{N68}$
92.500000	$2.276079\sigma^{N68}$	99.500000	$3.255247\sigma^{N68}$
95.000000	$2.447747\sigma^{N68}$	99.900000	$3.716922\sigma^{N68}$

As noted throughout the present section, a Rayleigh distribution can be formed as the sum of Gaussian distributed random variables. Table 2-14 indicates that a 1σ Gaussian random variable represents roughly 68% of all possible Gaussian outcomes, while Table 2-16 indicates that the 1σ containment level for the corresponding Rayleigh distribution represents roughly 39% of all possible Rayleigh outcomes. This relationship is presented in (2-64)

$$\sigma^{R39} = 1.0\sigma^{N68} \qquad \textbf{(2-64)}$$

Conversion from the underlying Gaussian distribution to a Rayleigh distribution with a specified containment level can be achieved by selecting the appropriate constant, *n*, from Table 2-16 or Table 2-17. For example, the conversion from a 1σ normally distributed random variable to a 95% Rayleigh random variable is given in (2-65). Figure 2-11 provides a Rayleigh scatter plot and probability circle enclosing 95% of all samples where the radius of the probability circle was determined from using (2-65).

$$\sigma^{R95} = 2.447747\sigma^{N68} \qquad \textbf{(2-65)}$$

Conversion between Rayleigh containment levels can be achieved by adjusting the given containment value to the 1σ Gaussian level, then converting to the desired Rayleigh containment level. For example, (2-66) demonstrates the process to convert from a Rayleigh 90% containment level to a Rayleigh 99% containment level using the values available from Table 2-17.

$$\sigma^{R90} = 2.145966\sigma^{N68} \Rightarrow \sigma^{N68} = \frac{\sigma^{R90}}{2.145966} \tag{2-66}$$

$$\sigma^{R99} = 3.034854(\sigma^{N68}) \Rightarrow 3.034854\left(\frac{\sigma^{R90}}{2.145966}\right) \Rightarrow \frac{3.034854}{2.145966}\sigma^{R90}$$

The preceding analysis concerning Rayleigh conversions could have also been performed using the Chi-Squared distribution as an intermediate step in the conversion process. This is because the Rayleigh distribution is a special case of the Chi distribution with two degrees of freedom. As a special case of the Chi distribution, by extension, the Rayleigh distribution is also a special case of the Chi-Squared distribution. Frank van Diggelen performed such an analysis in [31] and derived conversion constants equivalent to those presented in Table 2-17.

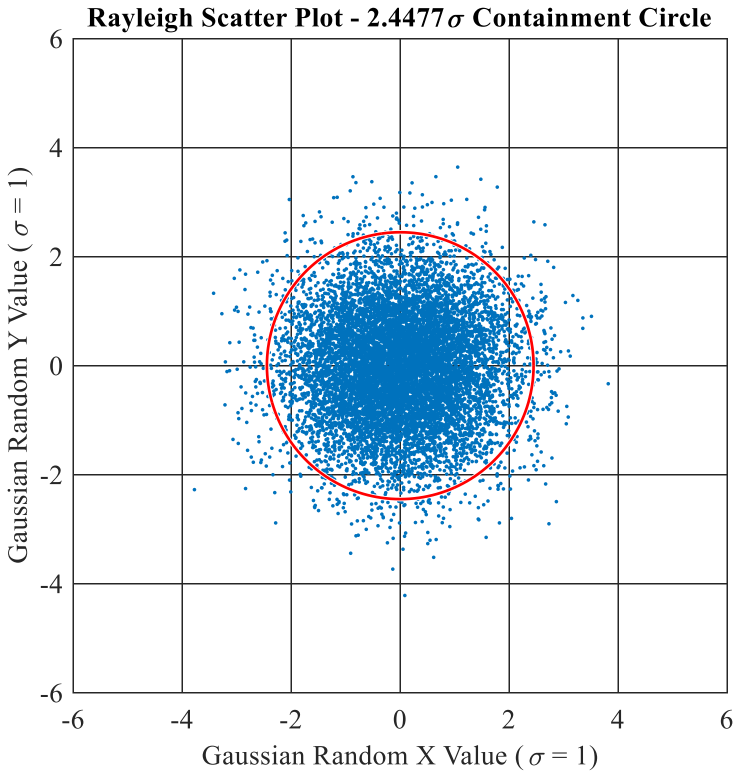

Figure 2-11: Rayleigh Random Variable Scatter Plot with 95% Containment Circle.

2.6 Earth Models and the Equations of Geodesic Paths and Positioning

The purpose of most navigation applications is to compute a position referenced to the Earth's surface in order to control a vehicle's movement from one position to another [32]. Fundamental to this concept is determining the vehicle's position relative to the Earth. This requires a model of the Earth to be developed that allows the vehicle's position to be expressed with respect to the model [32]. This modeling process is known as geodesy.

The Earth's true shape is that of an irregular oblate spheroid. That is, the Earth is nearly spherical in shape, but is slightly wider along the equator than it is along its axis of rotation, with the equatorial radius being 0.3% longer than the polar radius [32]. Furthermore, the surface of the Earth is irregularly shaped, making hyper-precise modeling impractical for most applications. Therefore, the surface is approximated to a regular shape, then fit to mean sea level as illustrated in Figure 2-12 [32].

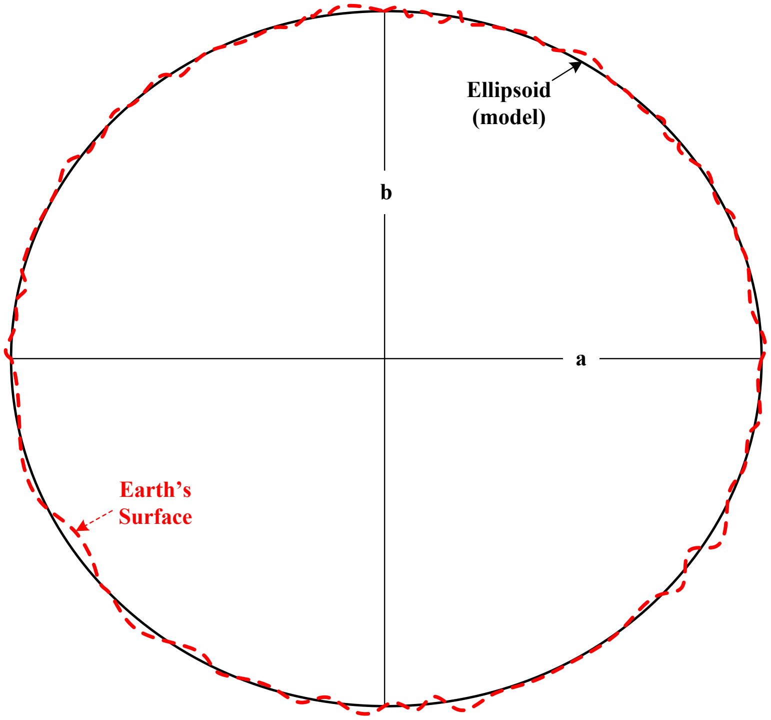

Figure 2-12: Ellipsoid fit to the Earth's Surface.

Some geodesic techniques consider the oblate nature of the spheroid, while others assume a strictly spherical representation. The choice of model largely depends on the required accuracy and computational capacity of the system performing the navigation calculations. In either case, two of the main challenges associated with geodesy are the calculation of the direct (sometimes referred to as forward) and inverse problems.

The direct problem seeks to determine a new position given a starting position, initial direction of travel, and distance of travel along the surface of the Earth. Conversely, the inverse problem seeks to determine the initial direction of travel and the shortest surface distance between two known positions. In each of these cases, the distance is expressed in terms of the great circle distance.

Geometrically, a great circle is a plane that passes through the center of the sphere (or spheroid) and cuts the surface into a circle (or oval in the case of a spheroid) [33]. Practically, the great circle distance is the shortest path between two points along the surface of the sphere, rather than through the sphere. As such, for any two non-antipodal points on the sphere, there is a single unique great circle, with the shorter path between the points being defined as the great circle distance.

The remainder of this section provides an overview of two geodesic Earth models: the spherical Earth model and the ellipsoidal model.

*2.6.1 **Spherical Earth Model***

The spherical Earth model disregards the flattening of the Earth and assumes that the Earth is a perfect sphere. This spherical assumption is less accurate than ellipsoidal models, but requires little more than an understanding of spherical trigonometry or vector algebra to perform calculations over the surface.

2.6.1.1 *Coordinate Frames and Transformations*

There are numerous coordinate frames that can be used to perform spherical Earth analysis, with the choice of coordinate system being dependent on the type of analysis being performed. This section describes some of the fundamental coordinate systems and establishes the relationship between them.

2.6.1.1.1 *Earth Centered, Earth Fixed Coordinate Frame*

Geocentric coordinate systems are often utilized in the spherical Earth model. A geocentric coordinate system is a Cartesian system with the origin at the center of the spherical model. The Z-axis is aligned with the polar axis, while the X-Y plane is situated in the equatorial plane [34]. The X-axis is aligned and fixed to the prime meridian; while the Y-axis completes the right handed coordinate system [34]. This rotating coordinate system is often referred to as Earth Centered, Earth Fixed (ECEF). An example of a spherical Earth model and the associated geocentric ECEF coordinate system are presented in Figure 2-13.

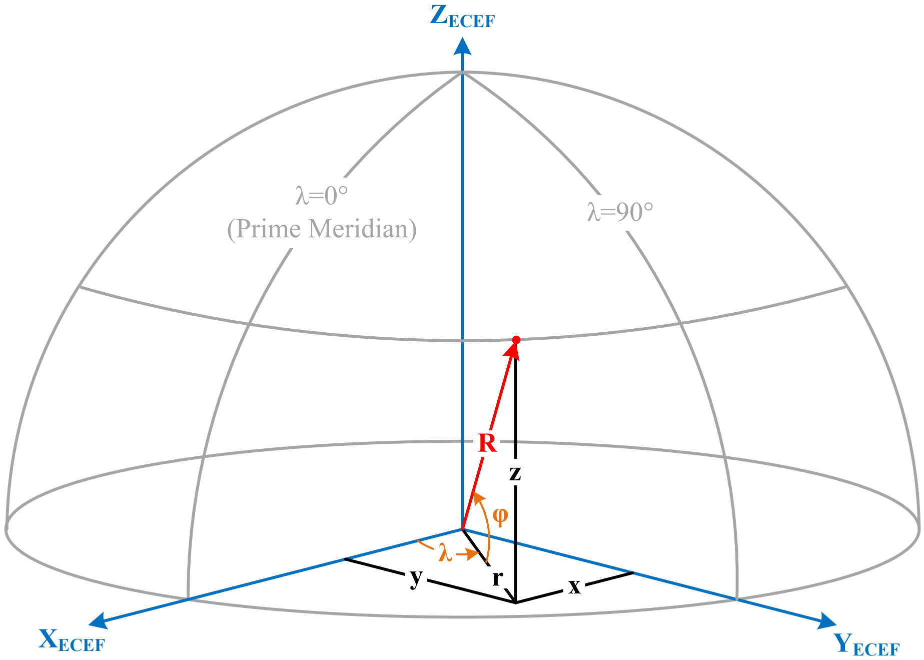

Figure 2-13: Spherical Earth Model and Geocentric Coordinate Frame.

2.6.1.1.2 *Geographic Coordinates*

Positions relative to the spherical Earth model can be expressed directly in the Cartesian ECEF frame, but are often expressed in terms of geographical latitude-longitude coordinates. Geocentric latitude, denoted as φ in Figure 2-13, is the vertical angle measured up from the equatorial plane to the point of interest on or above the sphere. The usual convention is that latitude is measured as positive in the northern hemisphere. This angle differs from the polar angle used in a traditional spherical coordinate systems because it is measured up from the X-Y plane rather than down from the Z-axis. Longitude, denoted λ, is equivalent to the azimuth angle in a traditional spherical system, and is measured in the equatorial plane from the X-axis (prime meridian) to the point of interest. Historically, longitude is positive to the east of the prime meridian.

2.6.1.1.3 *Local Geodetic Coordinate Systems*

A local geodetic coordinate frame, also known as a local level navigation frame or local tangent plane, is a coordinate system with an arbitrary origin that is typically coincident with the center of mass of the host vehicle [32]. The axes of the coordinate frame are often aligned with the cardinal directions, and with the Z-axis normal to the surface [32]. Two of the most common local geodetic frames are the North, East, Down (NED) frame and the East, North, Up (ENU) frame [32].

In the NED frame, the x-axis is aligned to North, the y-axis is aligned to East, and the z-axis points down to the center of the Earth to form an orthogonal right-handed coordinate system [32]. Alternatively, the ENU frame has the x-axis aligned to East, the y-axis aligned to North, and the z-axis pointing up and normal to the sphere to complete the right handed coordinate system [32]. Examples of the NED and ENU coordinate frames are given in Figure 2-14 and Figure 2-15 respectively.

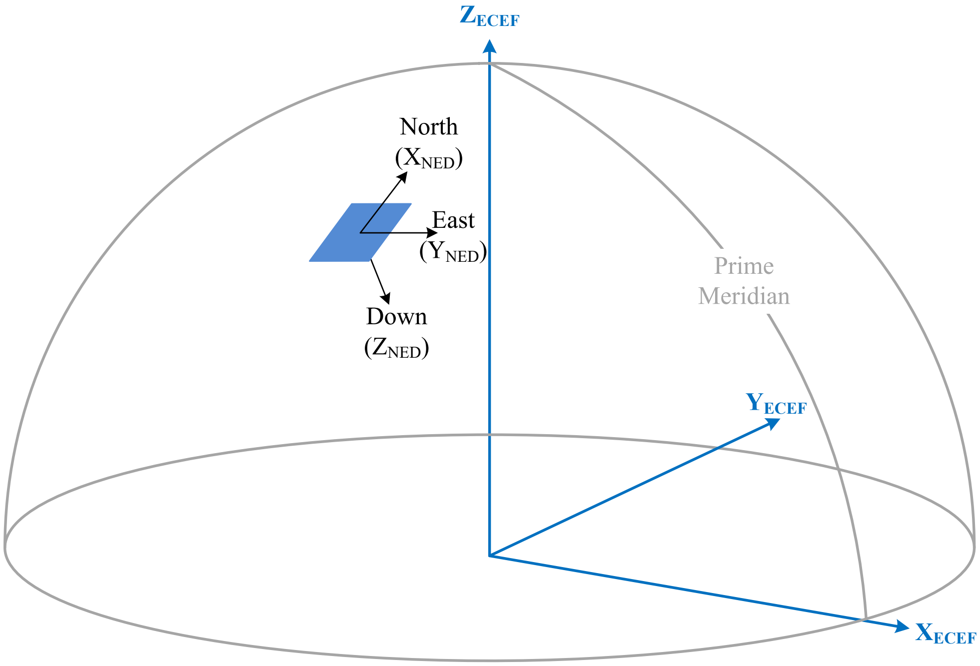

Figure 2-14: North, East, Down Local Geodetic Coordinate Frame.

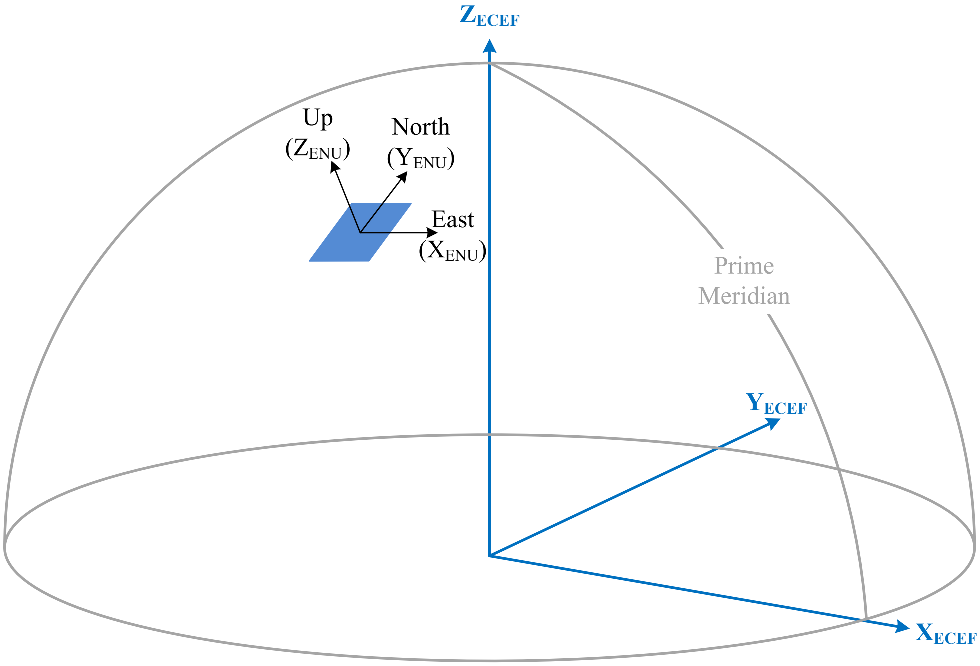

Figure 2-15: East, North, Up Local Geodetic Coordinate Frame.

2.6.1.1.4 *Earth Centered, Earth Fixed to Geographic Coordinate Frame Transformation*

Transforming from ECEF coordinates to geographical coordinates closely resembles Cartesian to spherical conversions and can easily be derived from Figure 2-13 as shown in (2-67) and (2-68). The use of a four quadrant arctangent function is recommended to prevent quadrant ambiguity.

$$sin(\varphi) = \frac{z}{R} \Rightarrow \varphi = arcsin\left(\frac{z}{R}\right) \qquad \textbf{(2-67)}$$

Where: φ = Geocentric latitude
R = Spherical radius of the Earth

$$tan(\lambda) = \frac{y}{x} \Rightarrow \lambda = arctan\left(\frac{y}{x}\right) \qquad \textbf{(2-68)}$$

Where: λ = Longitude

Likewise, conversion from geographical to ECEF coordinates can also be derived directly from Figure 2-13. This conversion is provided in (2-69) through (2-72).

$$r = (R)cos(\varphi) \qquad \textbf{(2-69)}$$

$$x = (r)cos(\lambda) \Rightarrow (R)\, cos(\varphi)\, cos(\lambda) \qquad \textbf{(2-70)}$$

$$y = (r)\, sin(\lambda) \Rightarrow (R)\, cos(\varphi)\, sin(\lambda) \qquad \textbf{(2-71)}$$

$$z = (R)sin(\varphi) \qquad \textbf{(2-72)}$$

The fundamental relationship between the geographical and ECEF coordinate systems enable spherical Earth calculations to be performed using either. The geographical coordinate system most readily lends itself to spherical trigonometric analysis; whereas, a vector approach is most commonly employed for analysis in the ECEF system.

2.6.1.1.5 *Earth Centered, Earth Fixed to Local Geodetic Transformations*

Conversion from ECEF to a local geodetic coordinate frame can be accomplished through the use of a coordinate conversion matrix, sometimes called a rotation matrix. The coordinate conversion matrix is an orthogonal matrix that is a function of the latitude and longitude of the origin of the local geodetic coordinate frame [32]. Once established, the coordinate conversion matrix is multiplied by the ECEF position vector, as measured from the local geodetic coordinate frame's origin to the position of interest. This operation simultaneously rotates and translates the position vector from the ECEF frame to the local geodetic frame. The conversion from ECEF to NED is expressed in (2-73) [32]; while (2-74) gives the conversion from ECEF to ENU [35]. The interested reader is directed to APPENDIX B and APPENDIX C for derivations of these coordinate conversion matrices.

$$\begin{aligned}
&\boldsymbol{r}_{NED} = \boldsymbol{C}_{ECEF}^{NED}\Delta\boldsymbol{r}_{ECEF} \\
&\Rightarrow \boldsymbol{r}_{NED} = \boldsymbol{C}_{ECEF}^{NED}(\boldsymbol{r}_{ECEF} - \boldsymbol{r}_o) \\
&\Rightarrow \boldsymbol{r}_{NED} = \begin{bmatrix} r_N \\ r_E \\ r_D \end{bmatrix} = \begin{bmatrix} -sin(\varphi_o)\,cos(\lambda_o) & -sin(\varphi_o)\,sin(\lambda_o) & cos(\varphi_o) \\ -sin(\lambda_o) & cos(\lambda_o) & 0 \\ -cos(\varphi_o)\,cos(\lambda_o) & -cos(\varphi_o)\,sin(\lambda_o) & -sin(\varphi_o) \end{bmatrix} \begin{bmatrix} \Delta r_x \\ \Delta r_y \\ \Delta r_z \end{bmatrix}
\end{aligned} \tag{2-73}$$

Where: $\boldsymbol{r}_{NED}$ = Position of interest expressed in the NED frame.

$\boldsymbol{C}_{ECEF}^{NED}$ = Coordinate conversion matrix from ECEF to NED.

$\Delta\boldsymbol{r}_{ECEF}$ = ECEF vector from NED origin to position of interest.

φ_o = Latitude of the NED frame's origin.

λ_o = Longitude of the NED frame's origin.

$\boldsymbol{r}_{ECEF}$ = Position of interest in ECEF coordinates.

$\boldsymbol{r}_o$ = Origin of the NED frame in ECEF coordinates.

$$\begin{aligned}
&\boldsymbol{r}_{ENU} = \boldsymbol{C}_{ECEF}^{ENU}\Delta\boldsymbol{r}_{ECEF} \\
&\Rightarrow \boldsymbol{r}_{ENU} = \boldsymbol{C}_{ECEF}^{ENU}(\boldsymbol{r}_{ECEF} - \boldsymbol{r}_o) \\
&\Rightarrow \boldsymbol{r}_{ENU} = \begin{bmatrix} r_E \\ r_N \\ r_U \end{bmatrix} = \begin{bmatrix} -sin(\lambda_o) & cos(\lambda_o) & 0 \\ -cos(\lambda_o)\,sin(\varphi_o) & -sin(\lambda_o)\,sin(\varphi_o) & cos(\varphi_o) \\ cos(\lambda_o)\,cos(\varphi_o) & sin(\lambda_o)\,cos(\varphi_o) & sin(\varphi_o) \end{bmatrix} \begin{bmatrix} \Delta r_x \\ \Delta r_y \\ \Delta r_z \end{bmatrix}
\end{aligned} \tag{2-74}$$

Where: $\boldsymbol{r}_{ENU}$ = Position of interest expressed in the NED frame.

$\boldsymbol{C}_{ECEF}^{ENU}$ = Coordinate conversion matrix from ECEF to NED.

As an orthogonal matrix, the coordinate conversion matrix exhibits the well-known properties given in (2-75) and (2-76).

$$\boldsymbol{C}\boldsymbol{C}^T = \boldsymbol{C}^T\boldsymbol{C} = \boldsymbol{I} \tag{2-75}$$

$$\boldsymbol{C}^T = \boldsymbol{C}^{-1} \tag{2-76}$$

As a result, algebraic manipulation of (2-73) or (2-74) can be used to establish the coordinate conversion matrix required to perform the conversion from the local geodetic coordinate frame back to ECEF. This derivation, given in (2-77), highlights that the transpose of the coordinate conversion matrix is used to perform the opposite conversion. This result is extended to the ENU rotation in (2-78).

$$\begin{aligned} &\boldsymbol{r}_{NED} = \boldsymbol{C}_{ECEF}^{NED}\Delta\boldsymbol{r}_{ECEF} \\ &\Rightarrow \Delta\boldsymbol{r}_{ECEF} = (\boldsymbol{C}_{ECEF}^{NED})^{-1}(\boldsymbol{r}_{NED}) \\ &\Rightarrow \boldsymbol{r}_{ECEF} = (\boldsymbol{C}_{ECEF}^{NED})^{T}(\boldsymbol{r}_{NED} + \boldsymbol{r}_{o}) \\ &\therefore \boldsymbol{C}_{NED}^{ECEF} = (\boldsymbol{C}_{ECEF}^{NED})^{T} \end{aligned} \tag{2-77}$$

$$\boldsymbol{r}_{ECEF} = \left(\boldsymbol{C}_{ECEF}^{ENU}\right)^{T}\boldsymbol{r}_{ENU} + \boldsymbol{r}_{o} \tag{2-78}$$

The coordinate conversion techniques presented in this section thus far have been expressed in terms of position conversions. It is equally valid to utilize coordinate conversion matrices to transform velocity vectors. In this case, it is not necessary to perform the translation between coordinate frame origins because velocity vectors do not represent a physical location. Accordingly, velocity conversion from ECEF to NED is simply the product of the rotation matrix and the velocity vector as presented in (2-79).

$$\dot{\boldsymbol{r}}_{NED} = \boldsymbol{C}_{ECEF}^{NED}\dot{\boldsymbol{r}}_{ECEF} \tag{2-79}$$

Where: $\dot{\boldsymbol{r}}_{NED}$ = Velocity expressed in the NED frame.
$\boldsymbol{C}_{ECEF}^{NED}$ = Coordinate conversion matrix from ECEF to NED.
$\dot{\boldsymbol{r}}_{ECEF}$ = Velocity expressed in ECEF coordinates.

Coordinate conversion matrices can also be used to rotate covariance matrices from one coordinate frame to another; however, these rotations do not follow the paradigm for position or velocity rotation because doing so would violate the scaling and summation properties of variances. Consequently, an equation of the form of (2-80) is used to mechanize the rotation of a covariance matrix [29]. Additional rationale for this equation is provided in APPENDIX D.

$$\boldsymbol{P}_{NED} = (\boldsymbol{C}_{ECEF}^{NED})(\boldsymbol{P}_{ECEF})(\boldsymbol{C}_{ECEF}^{NED})^{T} \tag{2-80}$$

Where: $\boldsymbol{P}_{NED}$ = Resulting rotated 3 x 3 covariance matrix in the NED frame.

$\boldsymbol{P}_{ECEF}$ = 3 x 3 covariance matrix expressed in the ECEF frame.

$\boldsymbol{C}_{ECEF}^{NED}$ = Coordinate conversion matrix from ECEF to NED.

2.6.1.2 Spherical Trigonometric Analysis of the Spherical Earth Model

Spherical Earth geodesic problems expressed in the geographical coordinate system are most easily solved using spherical trigonometric analysis. The remainder of this subsection provides an overview of the spherical law of cosines, then presents the solutions to the geodesic direct, geodesic inverse, great circle intersection, and angle of elevation problems in terms of spherical trigonometric relationships.

2.6.1.2.1 Spherical Law of Cosines

The solutions to most spherical problems can be found using the various forms of the spherical law cosines. Given the long history and rich application of the spherical law of cosines, a derivation has not been included here; however, a full derivation can be found in chapter 1 of W.M. Smart's work in [33]. For the sake of completeness, a simple restatement of the purpose and relevant permutations of the spherical law of cosines is provided in the remainder of this subsection.

The spherical laws of sines and cosines define a relationship between the sides and angles of an arbitrary spherical triangle, like the one shown in Figure 2-16. The spherical law of sines is expressed in (2-81), while the various permutations of the spherical law of cosines for sides are given in (2-82) through (2-84). Finally, the spherical law of cosines for angles is expressed in (2-85) through (2-87).

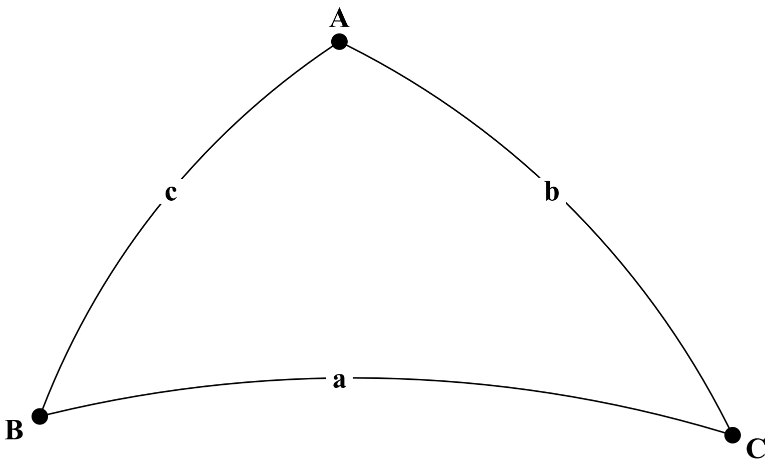

Figure 2-16: Arbitrary Spherical Triangle.

$$\frac{sin(a)}{sin(A)} = \frac{sin(b)}{sin(B)} = \frac{sin(c)}{sin(C)} \tag{2-81}$$

$$cos(a) = cos(b)\,cos(c) + sin(b)\,sin(c)\,cos(A) \tag{2-82}$$

$$cos(b) = cos(c)\,cos(a) + sin(c)\,sin(a)\,cos(B) \tag{2-83}$$

$$cos(c) = cos(a)\,cos(b) + sin(a)\,sin(b)\,cos(C) \tag{2-84}$$

$$cos(A) = sin(B)\,sin(C)\,cos(a) - cos(B)\,cos(C) \tag{2-85}$$

$$cos(B) = sin(C)\,sin(A)\,cos(b) - cos(C)\,cos(A) \tag{2-86}$$

$$cos(C) = sin(A)\,sin(B)\,cos(c) - cos(A)\,cos(B) \tag{2-87}$$

A set of noteworthy and useful corollaries to (2-82) through (2-84) allow an adjacent angle for a given side to be determined. Two such corollary equations exist for each form of the spherical law of cosines for sides, for a total of six corollary equations. For the sake of brevity, only the two relevant equations are presented here as (2-88) and (2-89).

$$sin(b)\,cos(A) = cos(a)\,sin(c) - sin(a)cos(c)\,cos(B) \tag{2-88}$$

$$sin(a)\,cos(B) = cos(b)\,sin(c) - sin(b)\,cos(c)cos(A) \tag{2-89}$$

W.M. Smart provides a derivation for the two corollary equations to determine the adjacent angles B and C of side a [33], one of which was given as (2-89). APPENDIX E expands this derivation to determine the adjacent angle A of side b as shown in (2-88).

Having established the fundamental spherical trigonometric equations in the present section, the remainder of Section 2.6.1 describes their application as related to navigation.

2.6.1.2.2 *Spherical Navigation Triangle*

For navigation purposes, a spherical triangle is defined by three vertices located on the surface of the sphere. One of the vertices is fixed at the North Pole; while the two remaining vertices represent the endpoints of a navigation leg [35]. Typically, the spatial relationship between the two endpoints is the focus of analysis. An example of an arbitrary navigation spherical triangle is provided as Figure 2-17, where the points P and Q represent the endpoints of the navigation leg, and the red line denotes the great circle path between them.

From Figure 2-17, it is evident that the great circle distance is related to the geocentric angle θ by the classic relationship shown in (2-90).

$$s = R\theta \tag{2-90}$$

Where: s = Great circle surface distance.

R = Radius of the sphere (Earth).

θ = Angle subtended by the great circle arc (geocentric angle).

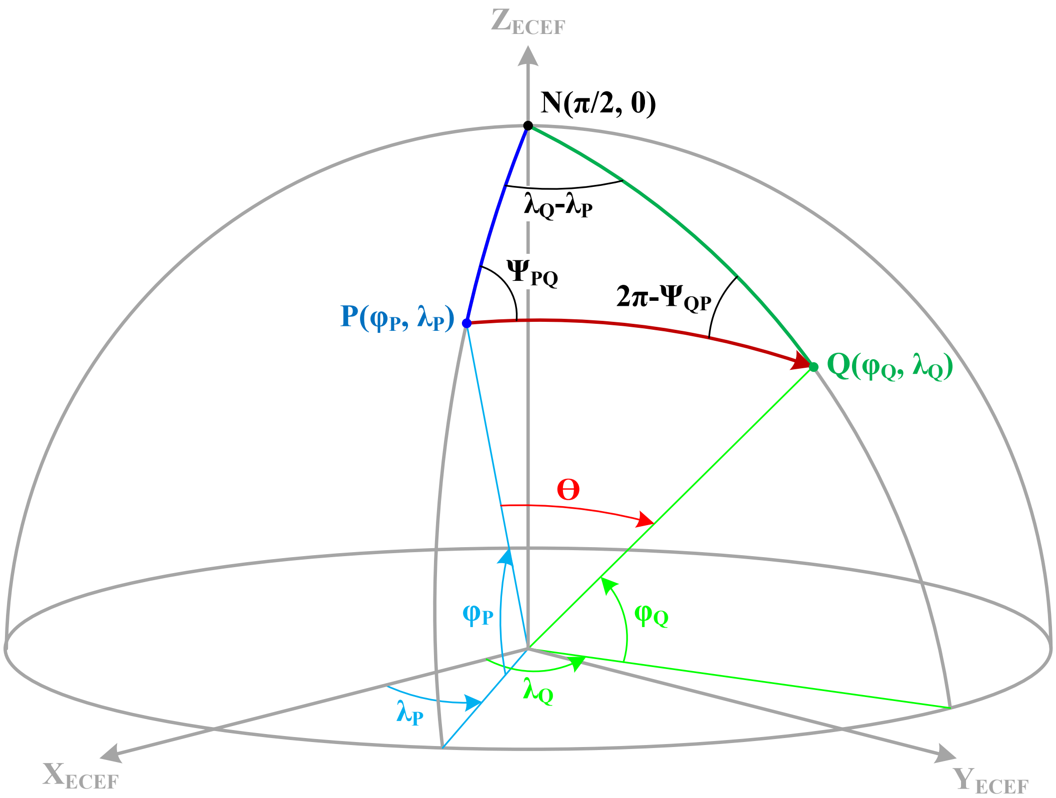

Figure 2-17: Spherical Navigation Triangle.

Likewise, the great circle distance from N to P is related to the co-latitude of P, and the great circle distance from N to Q is related to the co-latitude of Q in the same manner that θ is related to the great circle distance from P to Q. In some applications, the radius of the sphere is assumed to be unity, allowing the navigation calculations to be performed in terms of angular distances rather than surface distances. Once the problem has been solved in terms of angular distance, equation (2-90) can be used to express the result in terms of surface distance.

As discussed in Section 2.6.1.1.2, latitude is measured up from the equatorial plane; however, the spherical navigation triangle makes use of the corresponding polar angle, measured down from the Z-axis. This corresponding polar angle is known as the co-latitude; where the co-latitude is defined as the complimentary angle to the latitude $(90° - \varphi)$. A minor alteration to the formal definition of the spherical law of cosines can be made to allow the use of latitude directly, rather than co-latitude. This is done by replacing each trigonometric function that requires co-latitude with the appropriate latitude substitution listed in Table 2-18.

Table 2-18: Trigonometric Co-Function Substitutions for the Law of Cosines.

Function in terms of Co-Latitude	Function in terms of Latitude
$sin\left({}^{\pi}/_{2} - \varphi\right)$	$cos(\varphi)$
$cos\left({}^{\pi}/_{2} - \varphi\right)$	$sin(\varphi)$
$tan\left({}^{\pi}/_{2} - \varphi\right)$	$cot(\varphi)$

The spherical laws of sines and cosines expressed in terms of the spherical navigation triangle, and in terms of latitudes in lieu of co-latitudes, are given as (2-91) through (2-99).

$$\frac{sin(\theta)}{sin(\lambda_Q - \lambda_P)} = \frac{cos(\varphi_Q)}{sin(\Psi_{PQ})} = \frac{cos(\varphi_P)}{-sin(\Psi_{QP})} \tag{2-91}$$

$$cos(\theta) = sin(\varphi_Q)\,sin(\varphi_P) + cos(\varphi_Q)\,cos(\varphi_P)\,cos(\lambda_Q - \lambda_P) \tag{2-92}$$

$$sin(\varphi_Q) = sin(\varphi_P)\,cos(\theta) + cos(\varphi_P)\,sin(\theta)\,cos(\Psi_{PQ}) \tag{2-93}$$

$$sin(\varphi_P) = cos(\theta)\,sin(\varphi_Q) + sin(\theta)\,cos(\varphi_Q)\,cos(\Psi_{QP}) \tag{2-94}$$

$$cos(\lambda_Q - \lambda_P) = -sin(\Psi_{PQ})\,sin(\Psi_{QP})\,cos(\theta) - cos(\Psi_{PQ})\,cos(\Psi_{QP}) \tag{2-95}$$

$$cos(\Psi_{PQ}) = -sin(\Psi_{QP})\,sin(\lambda_Q - \lambda_P)\,sin(\varphi_Q) - cos(\Psi_{QP})\,cos(\lambda_Q - \lambda_P) \tag{2-96}$$

$$cos(\Psi_{QP}) = sin(\lambda_Q - \lambda_P)\,sin(\Psi_{PQ})\,sin(\varphi_P) - cos(\lambda_Q - \lambda_P)\,cos(\Psi_{PQ}) \tag{2-97}$$

$$cos(\varphi_Q)\,cos(\lambda_Q - \lambda_P) = cos(\theta)\,cos(\varphi_P) - sin(\theta)\,sin(\varphi_P)\,cos(\Psi_{PQ}) \tag{2-98}$$

$$sin(\theta)\,cos(\Psi_{PQ}) = sin(\varphi_Q)\,cos(\varphi_P) - cos(\varphi_Q)\,sin(\varphi_P)\,cos(\lambda_Q - \lambda_P) \tag{2-99}$$

2.6.1.2.3 Spherical Geodesic Direct Problem

In terms of the spherical navigation triangle presented as Figure 2-17, the direct problem seeks to determine the position $Q(\varphi_Q, \lambda_Q)$, given the position $P(\varphi_P, \lambda_P)$, the initial azimuth from P to Q (denoted Ψ_{PQ}), and the geocentric angle θ between P and Q. This is accomplished by determining the latitude and longitude of Q independently, with each being calculated via application of the spherical laws of sines and cosines.

2.6.1.2.3.1 Computing Spherical Geodesic Latitude

Solving for the latitude of point Q (φ_Q) in the spherical direct problem requires application of equation (2-93). Because latitude values are in the range of $[-\pi/2\,,\pi/2]$, application of the $arcsine$ function will not result in quadrant ambiguity, resulting in an unambiguous solution [35].

2.6.1.2.3.2 Computing Spherical Geodesic Longitude

Owing to the fact that none of the modified spherical law of cosines equations isolate the endpoint longitude, the longitude of the end point is calculated as the sum of the starting longitude and the longitude delta determined from the modified spherical law of cosines equations. However, care must be taken because the range of longitude values is $(-\pi,\pi]$; therefore, use of a four quadrant arctangent function is necessary to prevent quadrant ambiguity [35]. To that end, the solution for the longitude value in the direct problem requires an expression for both the sin and cos values of the longitude delta. To determine the sin of the longitude delta, the spherical law of sines, expressed in (2-91) is applied. This is shown in (2-100) [35].

$$sin(\lambda_Q-\lambda_P)=\frac{sin(\theta)\,sin(\Psi_{PQ})}{cos(\varphi_Q)} \tag{2-100}$$

An expression for $cos(\lambda_q-\lambda_p)$ can be determined by rearranging the alternate form of the law of cosines previously given in (2-98). This is given in (2-101).

$$cos(\lambda_Q-\lambda_P)=\frac{cos(\theta)\cos(\varphi_P)-sin(\theta)\,sin(\varphi_P)\cos(\Psi_{PQ})}{\cos(\varphi_Q)} \tag{2-101}$$

Finally, the change in longitude can be determined as given in (2-102) [35].

$$tan(\lambda_Q-\lambda_P)=\frac{sin(\lambda_Q-\lambda_P)}{cos(\lambda_Q-\lambda_P)}=\frac{sin(\theta)\,sin(\Psi_{PQ})}{cos(\theta)\,cos(\varphi_P)-sin(\theta)sin(\varphi_P)\,cos(\Psi_{PQ})} \tag{2-102}$$

It is possible that the result of these calculations may fall outside of the range of valid longitude $(-\pi, \pi]$ [35]. This is easily remedied by taking advantage of the periodic nature of trigonometric functions and adding/subtracting 2π as required to bring the result back into the desired range [35].

It is also interesting to note that the final solution depends only on the given input parameters, and does not require a chaining of results to compute the longitude. That is, longitude can be computed independently from latitude using only the given inputs, without the need to generate intermediate results.

2.6.1.2.4 *Spherical Geodesic Inverse Problem*

With respect to the spherical navigation triangle shown in Figure 2-17, the inverse problem is used to determine the geocentric angle θ, and the initial azimuth Ψ_{PQ}, given $P(\varphi_P, \lambda_P)$ and $Q(\varphi_Q, \lambda_Q)$. Some applications of the inverse problem also determine the angle of arrival into point Q, denoted Ψ_{QP}, but the current discussion will omit this derivation due to its similarity to the initial azimuth. Again, the solution to the problem requires two independent calculations, the first determines the great circle distance and the second determines the departure azimuth.

2.6.1.2.4.1 *Computing Spherical Great Circle Distance*

Determining the great circle distance from point P to point Q once again involves application of the modified spherical law of cosines. In this calculation, the goal is to find the geocentric angle θ given $P(\varphi_P, \lambda_P)$ and $Q(\varphi_Q, \lambda_Q)$, which can be accomplished through application of (2-92). Once the geocentric angle is resolved, the surface distance is easily determined by (2-90). Because the range of the resulting angle θ is $[0, \pi]$, the resulting great circle distance is an unambiguous solution.

Historically, the spherical law of cosines equation has been considered numerically ill-conditioned for small values of θ, and was not recommended for implementation [35]. This occurs because the derivative of $cos(\theta)$ approaches zero as the function value approaches ±1. That is, changes in the angular input argument near the local minima or maxima result in progressively smaller changes to the function value, which could be truncated or rounded off by the calculating system [35]. In the event that the application is required to compute small distances, the equivalent alternative function shown in (2-103), called the haversine equation, can be used to overcome this computational constraint. A complete derivation of the haversine solution is provided in APPENDIX F.

$$hav(\theta) = hav(\varphi_P - \varphi_Q) + cos(\varphi_P)\, cos(\varphi_Q)\, hav(\lambda_Q - \lambda_P) \tag{2-103}$$

Where: $hav(\theta) = sin^2\left(\frac{\theta}{2}\right) = \frac{1 - cos(\theta)}{2}$

2.6.1.2.4.2 Computing Spherical Departure Azimuth

The range of the departure azimuth is $(-\pi, \pi]$, which again necessitates the use of a four-quadrant arctangent function to resolve quadrant ambiguities. An expression for the *sin* of Ψ_{PQ} can be determined through application of the spherical law of sines given in (2-91). This is restated in the desired form in (2-104)

$$sin(\Psi_{PQ}) = \frac{cos(\varphi_Q)\, sin(\lambda_Q - \lambda_P)}{sin(\theta)} \tag{2-104}$$

The expression for *cos* of Ψ_{PQ} is determined by algebraic manipulation of the corollary expression to the spherical law of cosines for sides given in (2-99). This is given in the desired form as (2-105).

$$cos(\Psi_{PQ}) = \frac{sin(\varphi_Q)\cos(\varphi_P) - \cos(\varphi_Q)\, sin(\varphi_P)\cos(\lambda_Q - \lambda_P)}{sin(\theta)} \tag{2-105}$$

Applying the expressions for *sin* and *cos* to the tangent function results in the desired expression for determining the departure azimuth as presented in (2-106).

$$tan(\Psi_{PQ}) = \frac{sin(\Psi_{PQ})}{cos(\Psi_{PQ})} = \frac{cos(\varphi_Q)\,sin(\lambda_Q - \lambda_P)}{sin(\varphi_Q)\,cos(\varphi_P) - cos(\varphi_Q)\,sin(\varphi_P)\,cos(\lambda_Q - \lambda_P)} \quad \textbf{(2-106)}$$

2.6.1.2.5 Spherical Great Circle Intersection

The spherical great circle intersection problem is a higher-order algorithm that seeks to determine the intersection of two radials; where each radial is defined by its starting point and direction of travel. The intersection problem is considered a higher-order algorithm because it requires the application of both fundamental geodetic equations: forward and inverse.

Figure 2-18 presents the taxonomy of the spherical intersection problem, and will serve as a useful reference for the remainder of the present discussion. In terms of Figure 2-18, the goal of the spherical intersection problem is to determine $R(\varphi_R, \lambda_R)$ given: $P(\varphi_P, \lambda_P)$, $Q(\varphi_Q, \lambda_Q)$, the azimuth from P to R (Ψ_{PR}), and the azimuth from Q to R (Ψ_{QR}) [36].

The point of intersection, $R(\varphi_R, \lambda_R)$, can be found by solving the spherical direct problem detailed in Section 2.6.1.2.3 [36]. The inputs to the direct equations are the point $P(\varphi_P, \lambda_P)$, the azimuth from P to R (Ψ_{PR}), and the geocentric angle from P to R (θ_{PR}). Point P and azimuth Ψ_{PR} are given as inputs to the intersection problem; however, the geocentric angle θ_{PR} must be computed using the (2-87) instance of the spherical law of cosines before a solution to the direct problem can be determined. This is expressed in terms of Figure 2-18 in (2-107). Equations (2-108) through (2-110) define the parameters α, β, and γ of (2-107), where γ is an application of (2-85) [36].

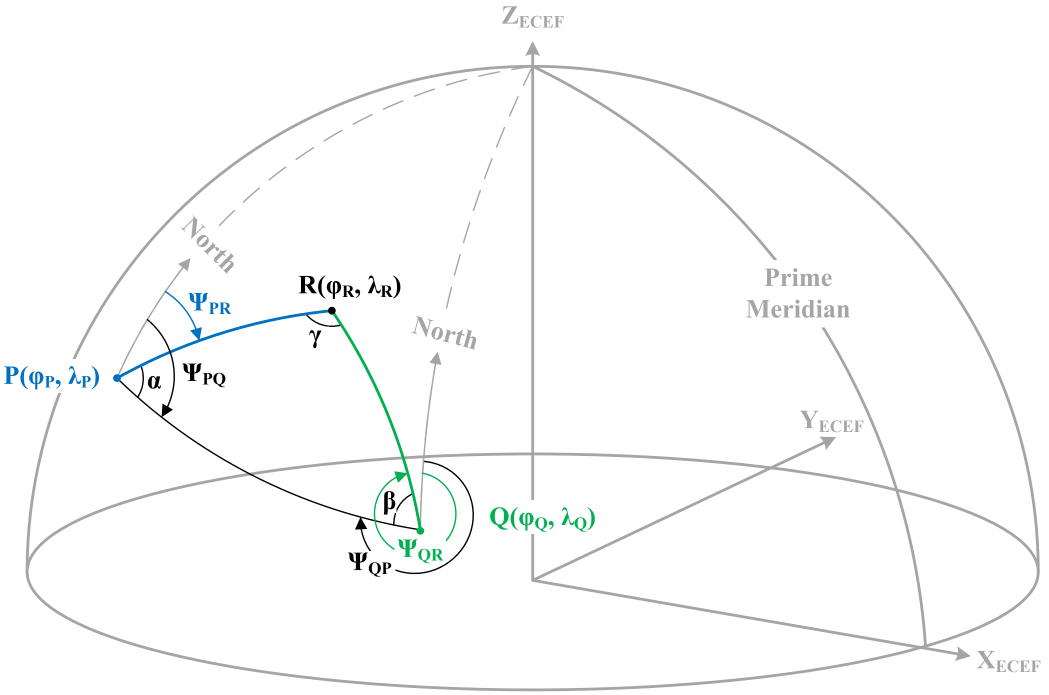

Figure 2-18: Taxonomy of Spherical Intersection Problem.

$$cos(\beta) = sin(\alpha)\, sin(\gamma)\, cos(\theta_{PR}) - cos(\alpha)\, cos(\beta) \tag{2-107}$$

$$\Rightarrow \theta_{PR} = \cos^{-1}\left[\frac{cos(\beta) + cos(\alpha)\, cos(\beta)}{sin(\alpha)\, sin(\gamma)}\right]$$

$$\alpha = \Psi_{PQ} - \Psi_{PR} \tag{2-108}$$

Where: α = Interior angle at point *P*, limited to the range (-*π*, *π*).

Ψ_{PQ}= Azimuth from *P* to *Q*, computed using spherical inverse.

Ψ_{PR}= Azimuth from *P* to *R*, given as input.

$$\beta = \Psi_{QR} - \Psi_{QP} \tag{2-109}$$

Where: β = Interior angle at point *Q*, limited to the range (-*π*, *π*).

Ψ_{QR}= Azimuth from *Q* to *R*, given as input.

Ψ_{QP}= Azimuth from *Q* to *P*, computed using spherical inverse.

$$\gamma = cos^{-1}\left[sin(\alpha)\, sin(\beta)\, cos\left(\theta_{PQ}\right) - cos(\alpha) cos(\beta)\right] \tag{2-110}$$

Where: γ = Interior angle at point *R*, limited to the range (-*π*, *π*).

θ_{PQ} = Geocentric angle from *P* to *Q*, computed with spherical inverse.

Care must be taken when computing the angles α and β to ensure that the resulting angles describe the internal angles of the spherical triangle. As such, (2-108) and (2-109) may require adjustment by $\pm 2\pi$ to ensure that the correct angles are used in the calculation.

As a higher-order algorithm, there may be cases when the result is undefined or ambiguous. For example, as the angles α and β both approach zero, the point of intersection converges to the line connecting P and Q. In this case, any point on the line PQ could be considered a point of intersection, resulting in an infinite number of solutions. Similarly, if the vectors originating at point P and Q are parallel, or nearly so, then either no intersection will exist, or the resulting intersection would require multiple circumnavigations of the sphere. Both cases should be considered invalid.

2.6.1.2.6 Spherical Altitude Given Elevation Angle

The equations presented in this section seek to determine the height of point P (h_P) above the reference ellipsoid, given the three-dimensional position of point Q and the angle of elevation β between the two points. The nomenclature for this geometric relationship is presented in Figure 2-19. As shown in (2-111), the planar law of sines can be used to solve for many of the indicated parameters.

$$\frac{sin(\theta)}{d_{PQ}} = \frac{sin(\gamma)}{R_m + h_Q} = \frac{sin(\eta)}{R_m + h_P} \tag{2-111}$$

Where: θ = Geocentric angle between points P and Q.

R_m = Mean radius of the Earth.

d_{PQ} = Slant range distance from P to Q.

h_P = Height (altitude) of point P above the sphere.

h_Q = Height (altitude) of point Q above the sphere.

$\gamma = \frac{\pi}{2} + \beta$

β = Angle of elevation.

$\eta = \pi - \gamma - \theta \Rightarrow \frac{\pi}{2} - \beta - \theta$

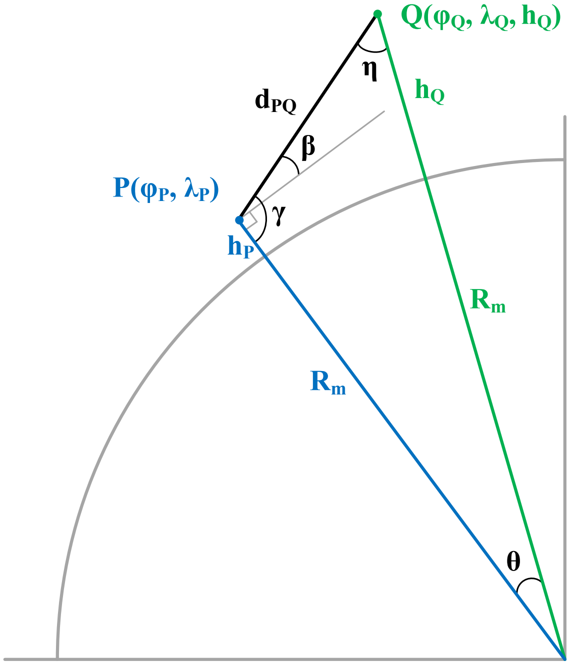

Figure 2-19: Nomenclature for Angle of Elevation Relationships.

Substituting for the angles γ and η in terms of β leads to the restatement of (2-111) given in (2-112).

$$\frac{sin(\theta)}{d} = \frac{cos(\beta)}{R_m + h_Q} = \frac{cos(\beta + \theta)}{R_m + h_P} \tag{2-112}$$

Algebraic manipulation of (2-112) allows the height of point P to be determined from the angle of elevation, two dimensional position of P, and three-dimensional position of Q. This is shown in (2-113). Note that the lateral position of each point is required to compute the geocentric angle using the spherical inverse calculations described in Section 2.6.1.2.4.1.

$$h_P = \frac{(R_m + h_Q)cos(\beta + \theta)}{cos(\beta)} - R_m \tag{2-113}$$

2.6.2 Ellipsoidal Earth Model

A more accurate model of the Earth can be formed by rotating an ellipse about its minor axis. This process forms an ellipsoid of revolution known as an oblate spheroid, and it serves as the basis for ellipsoidal Earth models used in most navigation systems [32]. An oblate spheroid is defined by two parameters, the semi-major axis a (also known as the equatorial radius R_0), and the semi-minor axis b (also known as the polar radius R_P). These defining parameters are illustrated in Figure 2-20 [34]. The oblate spheroid demonstrates rotational symmetry about the semi-minor axis and mirror symmetry about the semi-major axis [32].

Given the values of the semi-major and semi-minor axes, the flattening and eccentricity of the spheroid can be expressed as shown in (2-114) and (2-115) respectively [34].

$$f = \frac{a-b}{a} = 1 - \frac{b}{a} \tag{2-114}$$

$$e^2 = \frac{a^2 - b^2}{a^2} \tag{2-115}$$

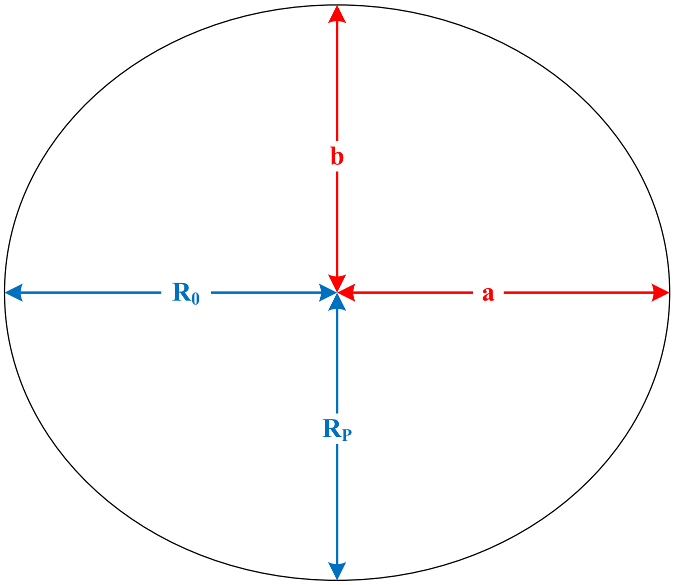

Figure 2-20: Ellipsoid Defining Parameters.

The semi-minor axis, flattening, and eccentricity are related by (2-116) and (2-117) [34].

$$e^2 = 2f - f^2 \tag{2-116}$$

$$\sqrt{1-e^2} = 1 - f = \frac{b}{a} \tag{2-117}$$

Using these relationships, the ellipsoid can be completely defined using the set of parameters a and b, or a and f, or a and e [34].

Like the spherical Earth model, an ECEF coordinate frame can be used to define a unique position on or above the ellipsoid. Also similar to the spherical Earth model, positions can be expressed in terms of latitude-longitude pairs, albeit they are geodetic rather than geocentric.

Geodetic longitude is measured in the same manner as geocentric longitude, making the two values equivalent. However, unlike the spherical Earth model, geodetic latitude values are not necessarily measured from the origin of the coordinate system. Geodetic latitude, Φ, is measured as the angle of intersection between the normal to the ellipsoid and the semi-major axis. This is illustrated in Figure 2-21 [32].

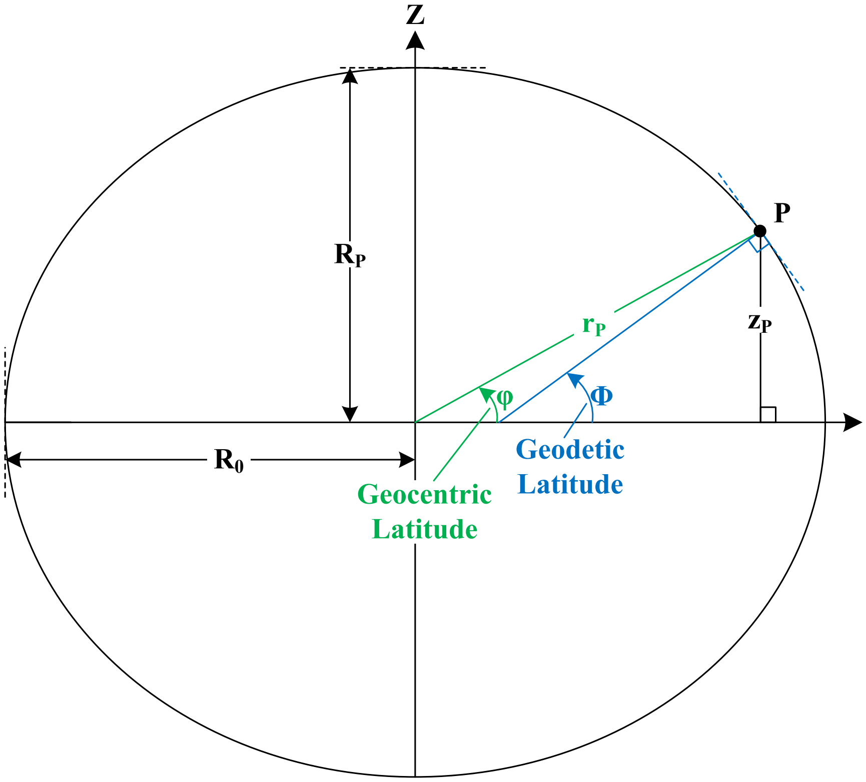

Figure 2-21: Geocentric Versus Geodetic Latitude.

In terms of ECEF coordinates, the geocentric latitude can be found using (2-118) [32].

$$tan(\varphi) = \frac{z_P}{\beta_P} = \frac{z_P}{\sqrt{(x_P)^2 + (y_P)^2}} \quad \textbf{(2-118)}$$

The relationship between geocentric and geodetic latitude is given in (2-119). A complete derivation of this relationship is provided by Groves in [32].

$$tan(\varphi) = (1 - e^2)tan(\Phi) \qquad \textbf{(2-119)}$$

The radius of curvature of the ellipse for north-south motion, known as the meridian radius of curvature, is a function of geodetic latitude and is given in (2-120) [32].

$$R_N = \frac{R_0(1 - e^2)}{(1 - e^2\, sin^2\, \Phi)^{3/2}} \qquad \textbf{(2-120)}$$

Likewise, the radius of curvature of the ellipse for east-west motion, known as the transverse radius of curvature, is also a function of geodetic latitude as shown in (2-121) [32].

$$R_E = \frac{R_0}{\sqrt{1 - e^2\, sin^2\, \Phi}} \qquad \textbf{(2-121)}$$

2.6.2.1 Geodetic to Earth Centered, Earth Fixed Transformations

Using similar geometry to the spherical Earth case, and the radius of curvature to account for the variable radius, (2-122) through (2-124) can be used to convert geodetic latitude-longitude coordinates into ECEF coordinates. A full derivation of this process is again provided by Groves in [32].

$$x = R_E\, cos(\Phi)\, cos(\lambda) \qquad \textbf{(2-122)}$$

$$y = R_E\, cos(\Phi)\, sin(\lambda) \qquad \textbf{(2-123)}$$

$$z = R_E(1 - e^2)\, sin(\Phi) \qquad \textbf{(2-124)}$$

2.6.2.2 Geodetic to Earth Centered, Earth Fixed Transformations

Conversion from ECEF to geodetic latitude requires iteration over (2-125). An iterative process is required because the transverse radius of curvature, R_E, used in the equation is itself a function of the resulting latitude [32].

$$tan(\Phi) = \frac{z(R_E)}{\sqrt{x^2 + y^2}[(1 - e^2)R_E]} \qquad \textbf{(2-125)}$$

Conversion from ECEF to geodetic longitude mirrors the conversion from ECEF to geocentric longitude given in Section 2.6.1. For completeness, the equation is restated as (2-126).

$$tan(\lambda) = \frac{y}{x} \tag{2-126}$$

2.6.2.3 Earth Centered, Earth Fixed to Local Geodetic Transformations

Conversion from ECEF to a local geodetic coordinate frame (NED or ENU) is accomplished in the same manner as the spherical transformation previously described in Section 2.6.1.1.5. For brevity, the equations are not repeated here. One noteworthy physical difference between spherical local geodetic coordinate frames and ellipsoidal local geodetic frames is that in the ellipsoidal case, the local frame is normal to the geodetic latitude and is tangent to the ellipse, but is not necessarily normal to the geocentric latitude that projects from the center of the Earth. This result is evident in Figure 2-21.

2.6.2.4 Ellipsoidal Direct and Inverse Problems

Solving the ellipsoidal direct and inverse problems is equivalent to solving the navigation triangle presented for the spherical case. However, the solution is complicated by the fact that the radius of the ellipse is not constant, rather it varies with latitude as noted in (2-120) and (2-121).

In his seminal work of 1975, Thaddeus Vincenty presented a compact iterative solution of the main geodesic problems that was suitable for implementation on desktop calculators of the time [37]. In most cases, the solutions required fewer than four iterations to converge with an error less than 0.5mm [37]. However, for nearly antipodal points, Vincenty's algorithm demonstrated slow convergence, or a complete failure to converge [38]. Despite this constraint, Vincenty's work was instrumental to the field of navigation because it provided a computationally efficient mechanism to solve geodesic problems over the ellipse; as such, it remains in wide use today. For briefness, the 12 equations comprising the direct solution and the

9 equations for the inverse solution have not been included here. The interested reader is encouraged to consult Vincenty's original work cited as [37].

A more modern solution to the main geodesic problems was developed by Karney in [38]. The Karney solution expanded Vincenty's work by increasing the accuracy of the solution to the order of 15 nanometers, while demonstrating convergence for all points, even those that are antipodal [38]. The increased accuracy and guaranteed convergence results in a more complex algorithm than the Vincenty solution; which, in turn, implies that the Karney algorithm requires more computation time than the Vincenty algorithm. Given the complexity of the Karney algorithm, the equations are not presented here, the interested reader is referred to [38].

2.6.2.5 WGS-84 Coordinate Reference System

The World Geodetic System 1984 (WGS84) was originally developed by the Defense Mapping Agency as the standard for U.S. military applications [32]. Its widespread use by GNSS and INS resulted in its global adoption as the standard for navigation systems [32].

WGS84 defines an ellipsoid and corresponding ECEF coordinate frame oriented at the Earth's center of mass [32]. WGS84 then serves as a model of the Earth's geoid for mapping and navigation applications [32]. The ellipsoid is defined by the equatorial radius and flattening; allowing the remaining parameters to be derived using (2-116) and (2-117). Table 2-19 lists the defining WGS84 parameters along with the resulting derived values.

Table 2-19: Defining Parameters for the WGS84 Ellipsoid [32].

Parameter	WGS84 Value
Equatorial Radius, R_0	6,378,137.0 m
Polar Radius, R_P	6,356,752.31425 m
Flattening, f	1/298.257223563
Eccentricity, e	0.0818191908425

Because WGS84 is defined as an ellipsoid of revolution, the entirety of the preceding ellipsoid Earth model discussion, and corresponding geodesic solutions, are applicable to the

WGS84 model, provided that the WGS84 defining parameters are utilized in the calculations. Likewise, any other ellipsoidal Earth model can be utilized by substituting the corresponding defining parameters. As a special case, the ellipsoidal equations can be used to solve the spherical Earth geodesic problems by letting $R_0 = R_P$. However, this is not recommended given the additional computational burden associated with the ellipsoidal equations.

2.6.3 Accuracy of the Spherical Earth Approximation

In an effort to empirically evaluate the accuracy of spherical Earth calculations, the great circle distance between numerous predefined points was computed using both spherical trigonometry and the Karney algorithm. As discussed in the introduction of Section 2.6, the irregular shape of the surface of the Earth makes accurate modeling of the Earth quite challenging. Therefore, the surface of the Earth is approximated to a regular shape and fit to the surface. As such, the accuracy of any geodesic calculation is limited to the accuracy of the corresponding Earth model. Bearing this constraint in mind, the assumption is made that the WGS84 Earth model is the de facto best fit and represents truth. References to accuracy expressed in the following analysis are measured with respect to the WGS84 ellipsoid using the Karney algorithm as truth.

Prior to analyzing the spherical Earth model for accuracy, a suitable value for the radius of the Earth must be determined. Obvious choices include the equatorial radius R_0 or the polar radius R_P. The choice of R_0 would result in very accurate measurements in the area of the equator, but would overestimate calculations performed away from the equatorial plane. Likewise, the choice of R_P would result in accurate north-south calculations, but would underestimate east-west calculations. Therefore, the most desirable value of the spherical Earth radius would be the value that minimizes the mean square of the error. This value is the mean of the semi-axes, given as (2-127) [35]. Applying WGS84 parameters leads to a mean radius of 6,371,008.7714 m. This value was used in the present analysis.

$$R_m = \frac{2a+b}{3} = \left(1 - \frac{1}{3}f\right)a \tag{2-127}$$

The difference between the spherical earth results and Karney results were plotted in terms of both the absolute error and relative error as shown in Figure 2-22 and Figure 2-23. These figures were generated by fixing the starting point at (latitude, longitude) = (0°, 0°), while the endpoint latitude was swept from -90° to +90° in steps of 2°, and the endpoint longitude was swept from -180° to +180° in 2° increments.

Figure 2-22 indicates that the maximum absolute error of 18.94 km occurred at the endpoint (48°, 0°). As expected, the errors are symmetric about the equatorial plane and the prime meridian.

Figure 2-23 illustrates that the maximum relative error of 0.561% occurred at the endpoint (2°, 0°). Again, the errors are symmetric about the equatorial plane and the prime meridian. This maximum relative error can be rationalized as the worst case relative error between the meridian radius of curvature at the starting latitude and the mean spherical radius; where the meridian radius of curvature was defined in (2-120). The worst case error occurs when the starting point is located at the equator and the direction of travel is north or south along a meridian. This relative error expression is given as (2-128).

$$\%Err = 100\left(\frac{\Delta R}{R_N}\right) = 100\left(\frac{|R_N - R_m|}{R_N}\right) \tag{2-128}$$

Solving for R_m and R_N using WGS84 parameters, and assuming a 0° latitude in the R_N calculation, leads to (2-129). This result confirms the empirical relative error analysis shown in Figure 2-23.

$$\%Err = 100\left(\frac{|6{,}335{,}439.327 - 6{,}371{,}008.771|}{6{,}335{,}439.327}\right) \cong 0.561\% \tag{2-129}$$

This relative error is the direct result of the spherical Earth model's failure to consider the non-constant radius, or flattening of the Earth. In spite of the relative error, the elementary nature of the spherical Earth model is well suited for navigation applications, especially those where an error of this magnitude can be tolerated or otherwise compensated for.

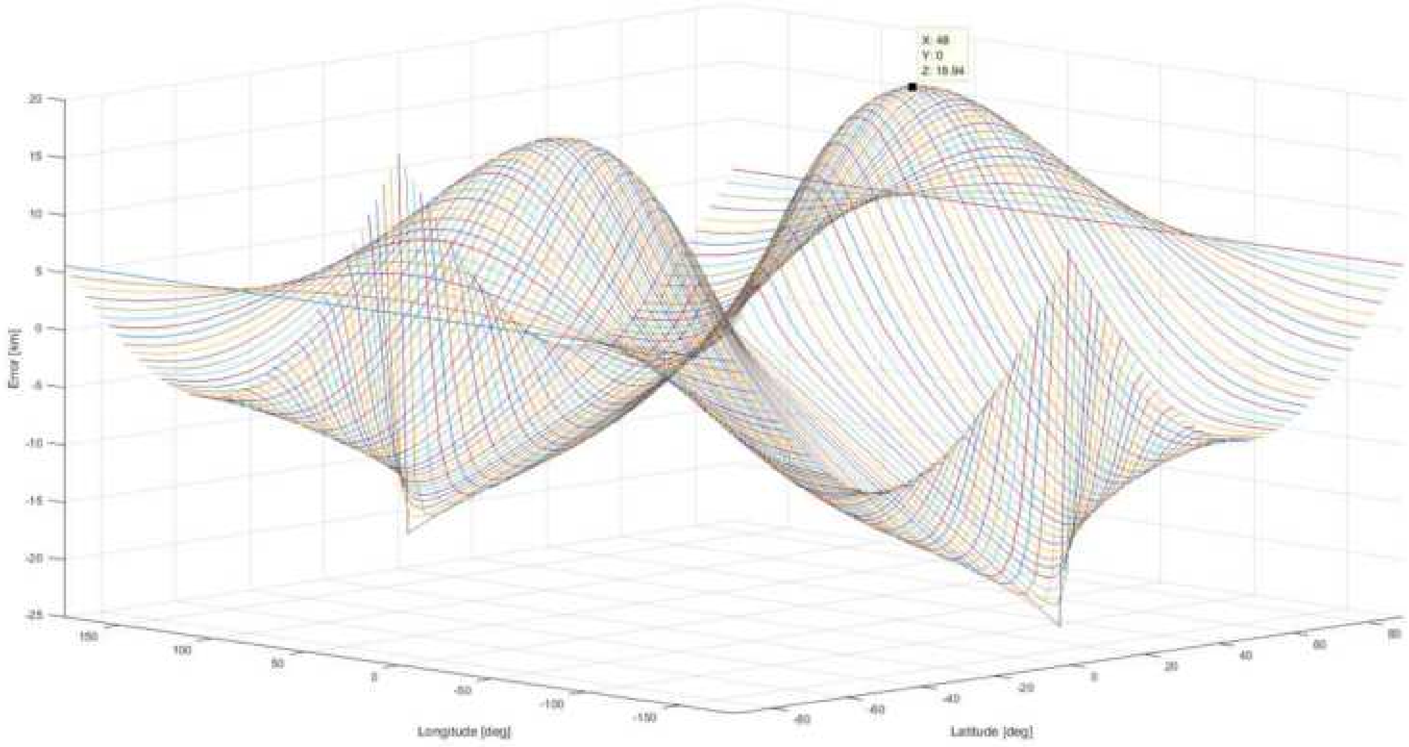

Figure 2-22: Absolute Error between the Spherical Earth Model and Karney Algorithm Over the range from (0°, 0°) to ([-90°, 90°], [-180°, 180°]).

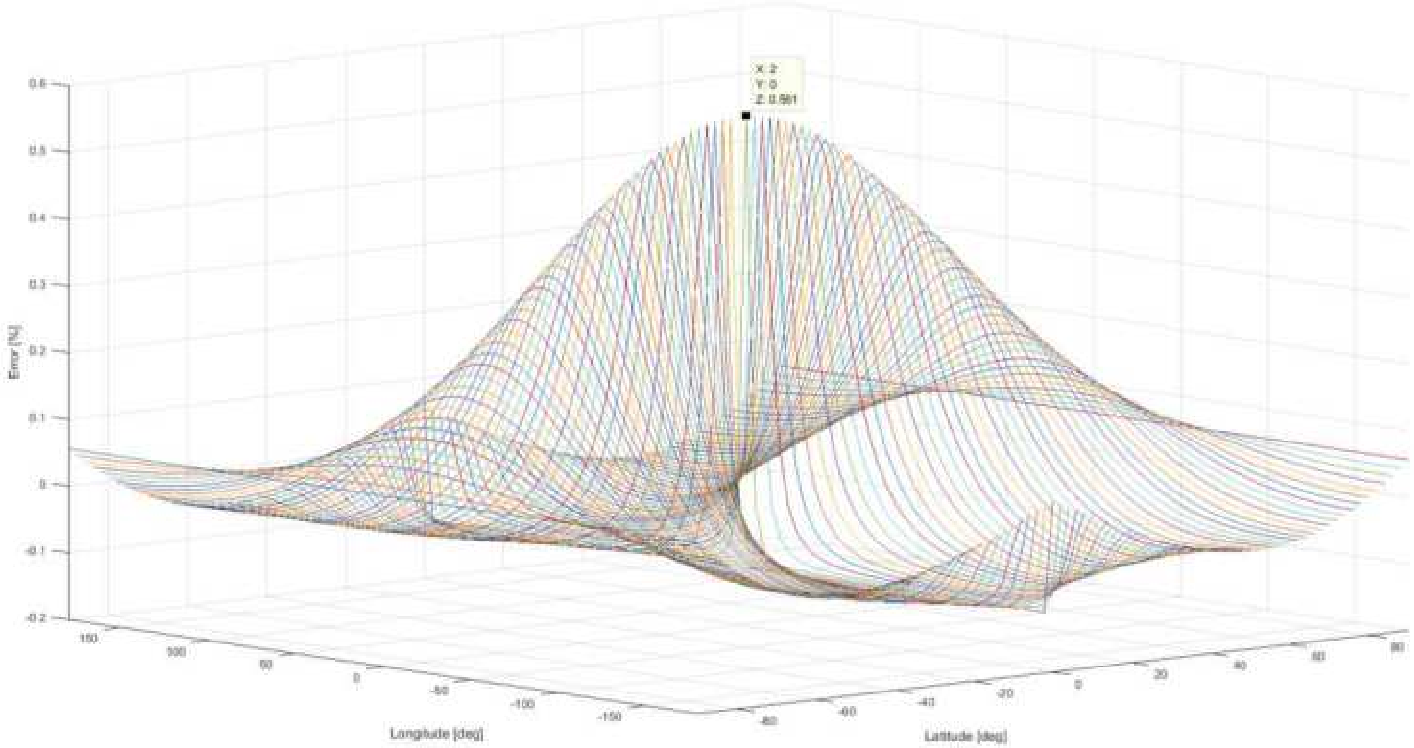

Figure 2-23: Relative Error between the Spherical Earth Model and Karney Algorithm Over the range from (0°, 0°) to ([-90°, 90°], [-180°, 180°]).

3.0 REVIEW OF LITERATURE

Although both ADS-B and direction finding are well understood and deployed in commercial products, there is relatively little existing research directly related to geolocation from ADS-B AOA measurements. Because a comprehensive review of ADS-B and its capabilities as defined by industry specifications was provided in Section 2.1, no additional literature was reviewed for this topic. Direction finding has a long history with a plethora of publications on the subject; however, few are directly related to ADS-B AOA. Because a summary of AOA technology was provided in Section 2.2, the AOA topics contained in this literature review are limited to those that relate to ADS-B AOA. In addition to ADS-B AOA, a few works relative to determining the uncertainty in geolocation based on direction finding will be reviewed, as these are somewhat applicable to the current research. Additionally, an assessment of three patents that were discovered whose embodiments contain aspects similar to the current research will be provided. Finally, a review of a published method to determine vehicle orientation from AOA/AOE measurements is included.

3.1 Angle-of-Arrival of Automatic Dependent Surveillance – Broadcast Messages

This section provides a review of pertinent publications that have investigated the practicality of determining the AOA of ADS-B transmissions. The most prominent research in this area was performed by Christoph Reck and his team, who contributed at least six papers on the topic. In addition to Reck, Faragher produced one publication on the subject of ADS-B AOA. The works by these authors are reviewed in the subsequent sub-sections.

3.1.1 Christoph Reck, et al.

Christoph Reck and his team of researchers have thoroughly studied the modern AOA techniques, reviewed in Section 2.2, with the specific intent to determine which algorithms are best suited for determining the AOA of ADS-B transmissions. In Reck's works, the ADS-B is received by precisely positioned ground based antenna arrays, consisting of either 6 or 8 elements [2][3][4][5][6][7].

Reck's team determined that the Matrix Pencil approach of AOA estimation was the least accurate due to its susceptibility to outliers; while the traditional ESPRIT and SVD ESPRIT algorithms were the most robust in the presence of imperfect hardware, carrier offsets, and noise [5]. Reck theorizes that a RMSE on the order of 0.3° is achievable using ESPRIT, and empirically demonstrated a RMSE of 1.0° using SVD ESPRIT when tested against 14,479 random ADS-B position samples generated from a random sample of ADS-B capable aircraft [5]. In these experiments, Reck compared the AOA of the received ADS-B transmission with the expected AOA; where the expected AOA was computed based on the receiving antenna location and the ADS-B position report [7]. This research demonstrated that the RMSE in AOA estimation remained at or below 1.0° for all ADS-B message types except for Types 17 and 18, where the RMSE increased to the order of 3° [6]. This result is justified by noting that ADS-B message Types 17 and 18 represent position transmissions that correspond to a radius of containment of 8 NM and 20 NM respectively (refer to Section 2.1.1.2.2). In these cases, the error in the reported ADS-B position resulted in an erroneous AOA estimate; thereby, increasing the perceived AOA error.

Reck's team postulated that because the contribution to the AOA error induced by ADS-B position error should decrease with increasing distance, the RMSE of the resulting AOA measurements should decrease over increasing distance as well [6]. Based on this theory, Reck et al., repeated their experiment, but limited the ADS-B data set to those signals that were Type 13 or less ($R_C < 0.5$ NM), were more than 30 km away, and whose angle of incidence on the antenna array was in the range 45° to 135° [6]. Reck's results were nearly piecewise linear, showing a RMSE of 0.9° for targets out to 75 km, a RMSE of 0.63° for targets at 135 km, and a RMSE below 0.4° for targets beyond 150 km [6].

One final embodiment of Reck's research utilized a dedicated ADS-B capable target aircraft to validate the accuracy of AOA estimates [7]. The target aircraft contained a highly accurate navigation sensor suite capable of estimating its position to the sub-meter accuracy level [7]. This ensured that the observed AOA errors were a function of the AOA measurement sensor and not induced by errors present in the transmitting aircraft's position report. This experiment empirically demonstrated that an AOA RMSE on the order of 0.66° could be achieved for aircraft operating at a variety of ranges [7]. This empirical result is likely the best indicator of the level of accuracy that can be realized from the ADS-B AOA estimation process using a stationary 8-element planar antenna array.

3.1.2 Ramsey Faragher, et al.

Faragher's research sought to develop an ADS-B AOA receiver to detect malicious attacks on the ADS-B network that may include the spoofing of GNSS, or the 'ghosting' of ADS-B data [8]. Ghosting is generally considered the process of simulating non-existent ADS-B aircraft with the intent of inspiring the collision avoidance algorithms in real aircraft to take evasive maneuvers to avoid the threat. Oftentimes the goal is to force the real aircraft into a flight path deviation, which may include an intentional crash into the ground.

Faragher primarily focused on phase interferometry techniques to determine the AOA of received ADS-B signals. To this end, he utilized a dual channel software defined radio with a two antenna system [8]. Each antenna was a simple two element array, with quarter wavelength diploes, and a basic ground plane reflector [8]. Faragher conducted two sub-experiments, one with the antennas separated by 14 cm (half wavelength), and the other with the antennas separated by 1.6 meters [8].

Faragher found that the setup was sensitive to errors in the antenna spacing, antenna orientation, and phase offset between receiver channels [8]. To compensate for these sensitivities, he developed a Bayesian estimation algorithm to characterize these errors [8]. Following calibration using the Bayesian estimator, Faragher observed an accuracy of 6° when 14 cm antenna spacing was used, and an accuracy of 1° when 1.6 meter antenna spacing was utilized [8].

3.2 Uncertainty in Direction Finding Implementations

Once operational, the algorithm will generate an estimate of the vehicle's position and the associated uncertainty in that estimate. However, a preliminary position uncertainty estimate is required to initialize the filter. To that end, two approaches for developing an estimate of lateral position uncertainty for AOA derived positions are reviewed.

3.2.1 Airborne Direction Finding – The Theory of Navigation Errors

Ancker published an approach to determine the uncertainty in a lateral position determined from the intersection multiple of angular bearings, where the bearings are measured with respect to airborne direction-finding stations [39]. These airborne direction-finding stations were assumed to have their own positional uncertainty, which increases the uncertainty in the resulting position fix [39].

Ancker used the variance of the AOA measurement, along with knowledge of each direction-finding station's position and variance, to construct a PDF that represents the statistical distribution of position errors generated when multiple lines of bearing are used to determine a position fix [39]. Generation of the PDF shown in (3-1) is subject to the following assumptions [39]:

1. The DF bearings have zero mean, normally distributed errors.
2. The DF bearings have small errors.

3. The navigation error is small compared to the range from the DF station to the position fix.
4. Navigation errors at each DF station are independent.
5. Navigation errors at each DF station have a circular frequency function.

$$f(L_j) = \frac{1}{\sqrt{2\pi}\,\sqrt{\sigma_{\psi_j}^2 D_j^2 + \sigma_j^2}} e^{\left[\left(-\frac{1}{2}\right)\left(\frac{L_j^2}{\sigma_{\psi_j}^2 + \sigma_j^2}\right)\right]} \qquad \textbf{(3-1)}$$

Where: $f(L_i)$= PDF of navigation errors as a function of DF station.

$\sigma_{\psi_j}^2$ = Variance of angle of arrival measurement to DF station j.

D_j^2 = Distance from position fix to DF station j.

σ_j^2 = Variance of position error in DF station j.

L_j^2 = Sum of the true bearing error and DF position error.

The variance along an arbitrary X and Y axis can then be expressed as given in (3-2).

$$(\sigma_X^{N68})^2 = \frac{\mu}{\lambda\mu - \nu^2} \qquad (\sigma_Y^{N68})^2 = \frac{\lambda}{\lambda\mu - \nu^2} \qquad \textbf{(3-2)}$$

Where:

$$\lambda = \sum_1^n \frac{cos^2(\alpha_j)}{\sigma_{\psi_j}^2 D_j^2 + \sigma_j^2}$$

$$\mu = \sum_1^n \frac{sin^2(\alpha_j)}{\sigma_{\psi_j}^2 D_j^2 + \sigma_j^2}$$

$$\nu = \sum_1^n \frac{sin(\alpha_j)\,cos(\alpha_j)}{\sigma_{\psi_j}^2 D_j^2 + \sigma_j^2}$$

α_j = True bearing angle to DF station j.

These variance values could be used to determine a distribution that represents the position uncertainty in the computed position. If for no other reason than computing the uncertainty of the initial position, this algorithm is applicable to the current research.

3.2.2 Dilution of Precision in Angle-of-Arrival Positioning Systems

Dilution of Precision (DOP) relates ranging (measurement) errors to the error in computed position [40]. It has also become common to use DOP to quantify the effect of the receiver-satellite geometry on the accuracy of a GNSS position [41]. The mathematical expression for DOP is given in (3-3).

$$\sigma_a = \mathrm{DOP}\sigma_r \tag{3-3}$$

Where: σ_a = Standard deviation of computed position.

σ_r = Standard deviation of range measurement.

In his work, Dempster derives a DOP factor for two-dimensional angle-of-arrival positioning systems based on the illustration given in Figure 3-1 [42]. The uncertainty in the AOA measurements for two stations can be related to the user position (x_u, y_u) uncertainty as shown in (3-4) [42].

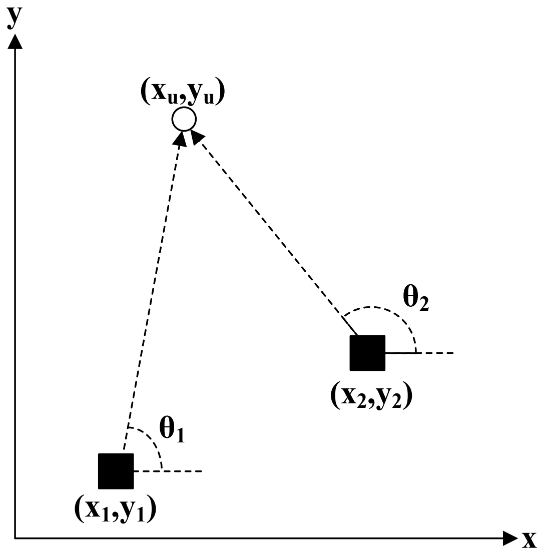

Figure 3-1: Geometry for Angle-of-Arrival Dilution of Precision Derivation [42].

$$\begin{bmatrix} d\theta_1 \\ d\theta_2 \end{bmatrix} = \boldsymbol{H} \begin{bmatrix} dx_u \\ dy_u \end{bmatrix} = \begin{bmatrix} \frac{-(y_u - y_1)}{r_1^2} & \frac{(x_u - x_1)}{r_1^2} \\ \frac{-(y_u - y_2)}{r_2^2} & \frac{(x_u - x_2)}{r_2^2} \end{bmatrix} \begin{bmatrix} dx_u \\ dy_u \end{bmatrix} \tag{3-4}$$

Where: $d\theta_n$ = Error in AOA from station *n*.

r_n = Range from station *n* to the user position.

x_n = Station *n* x-position.

y_n = Station *n* y-position.

x_u = User's x-position.

y_u = User's y-position.

dx_u = Error in user's x-position.

dy_u = Error in user's y-position.

The DOP value can then be computed using (3-5) [42].

$$\text{DOP} = \sqrt{trace[(\boldsymbol{H}^T\boldsymbol{H})^{-1}]} \tag{3-5}$$

Of more interest to the current research is the error in the user's position, where the user's position has been computed using AOA measurements. From (3-4), it is evident that the error in the user's position can be determined as given in (3-6).

$$\begin{bmatrix} dx_u \\ dy_u \end{bmatrix} = \boldsymbol{H}^{-\mathbf{1}} \begin{bmatrix} d\theta_1 \\ d\theta_2 \end{bmatrix} \tag{3-6}$$

For an over-specified system (more than two stations), additional rows can be added to $\boldsymbol{H}$ following the same form. Then, a least-squares approximation of the error in the user's position could be computed using a pseudo-inverse as shown in (3-7) [42].

$$\begin{bmatrix} dx_u \\ dy_u \end{bmatrix} = (\boldsymbol{H}^T\boldsymbol{H})^{-\mathbf{1}}\boldsymbol{H}^T \begin{bmatrix} d\theta_1 \\ d\theta_2 \\ . \\ . \\ d\theta_n \end{bmatrix} \tag{3-7}$$

3.3 Patented Work Related to Research

This section provides a brief overview of three patents related to ADS-B validation and provides an assessment of the relevance of each disclosure to the present research.

3.3.1 Device, System and Methods Using Angle of Arrival Measurements for ADS-B Authentication and Navigation

United States Patent 2014/0327581 A1 details a method for utilizing AOA measurements to authenticate ADS-B transmissions. In addition, it describes a method to utilize AOA from ground based targets of opportunity to determine the receiver's position [43]. The patent proposes that ADS-B authentication can be achieved by comparing the expected AOA derived from received ADS-B position and track with the actual angle of arrival of the RF signal determined using a direction finding receiver [43]. Discrepancies between the estimated AOA and the measured AOA would result in an indication of an ADS-B position validation failure.

This theory is directly applicable to the current research because this is the method used to generate the EFK residuals that generate the filter gains. Refer to Section 2.3.1, and Equation (2-23) for additional information concerning the implementation of this principle. It should be noted; however, that the use of residuals to gauge the validity of a measurement is not novel, and has been fundamental in Kalman Filter implementations since the 1960's. What may be considered novel is the use of ADS-B position information to calculate the estimated AOA.

A secondary embodiment of [43] is a means to determine the receiver's position using AOA information from multiple ground based targets of opportunity that transmit in the aviation frequency band [43]. This embodiment is also applicable to the current research as the present navigation solution utilizes multiple ADS-B AOA signals in a least-squares fashion to determine an estimate of the receiver's position. However, it must be noted that the cited patent limits the position determination algorithm to the use of ground based targets of opportunity, where the position of the transmitter is known. The current research utilizes airborne targets of opportunity to achieve a similar result. In the airborne case, the targets of opportunity are transmitting their position in the form of ADS-B packets, and the algorithm will simultaneously track and generate its own estimate of the target's position.

3.3.2 Validity Check of Vehicle Position Information

United States Patent 2011/1063908 A1 describes a method for validating the position information being transmitted by an ADS-B capable aircraft using a direction finding antenna to determine the bearing from the receiver to the transmitter [44]. Given the position of the receiver, the bearing to the transmitter, and the distance to the transmitter, the transmitter's location can be precisely determined and validated against the transmitted position [44]. The employment of the direction finding antenna to receive ADS-B data is directly applicable to the current research and provides additional confidence that such a device is feasible. However, the disclosure is currently impractical because the proposal theorizes that the distance to the transmitting aircraft can be calculated based on time of flight of the RF signal. This is not currently possible with ADS-B data packets because the time of transmission is not a member of the ADS-B data set. Therefore, an additional distance measuring sensor would be required to make this invention practical. Therefore, the concept is recognized as applicable to the current research, but of little tangible value.

3.3.3 System and Method for Ensuring ADS-B Integrity of Departing Aircraft

United States Patent 2015/9116240 B2 discloses an invention that seeks to validate the integrity of ADS-B data being transmitted by an aircraft preparing for departure [45]. The invention describes a system that would monitor the ADS-B output of an aircraft positioned in a designated area prior to departure. The invention would then provide a visual indication to the aircrew that the information being transmitted is valid and correct [45]. As this disclosure simply describes a means to validate ADS-B output data without relying on or deriving an AOA from the ADS-B data, the information contained within this disclosure is of little interest to the present research.

3.4 Three-Dimensional Self-Localization from Angle of Arrival Data

Kim and Hmam present a method for determining a vehicle's orientation and position based on three-dimensional AOA measurements made to stationary landmarks [46]. In this approach, the collinearity error between the estimated line-of-sight vector and the measured line-of-sight vector is minimized to develop the estimate [46]. Kim and Hmam claim that the algorithm displays rapid convergence and tolerates large measurement and tilt errors [46].

Kim and Hmam's research is not directly applicable to the current investigation because the realistic assumption has been made for this research that the host vehicle contains a sensor capable of determining the vehicle's orientation; therefore, self-orientation is not required. Furthermore, Kim and Hmam's work relies on the use of stationary landmarks. However, because Kim and Hmam's research utilize AOA/AOE measurements in a similar manner to the current research, it may be plausible to extend their work to utilize airborne navigation aids instead of stationary landmarks.

4.0 METHOD

The method described in this paper implements an EKF-SLAM-like algorithm that tracks the position and velocity of airborne navigation aids while simultaneously calculating the geodetic position of the host vehicle. Instead of utilizing fixed landmarks typical in a SLAM implementation, ADS-B capable aircraft act as airborne navigation aids (SLAM landmarks). Given the proliferation of ADS-B capable aircraft, the accuracy and availability of ADS-B position data, and the ability to accurately measure ADS-B AOA, ADS-B capable aircraft are ideal candidates for use as airborne navigation aids for a geo-positioning algorithm like the one described in this document.

The navigation algorithm determines its best estimate of the position of each airborne navigation aid through fusion of the reported ADS-B information and the AOA/AOE of the received signal. Tracking each navigation aid independently allows the system to detect erroneous ADS-B position reports, and account for them, while utilizing the data to determine the position of the host vehicle. Detection of erroneous ADS-B position reports could then be forwarded to other aircraft and ADS-B ground stations, although this is considered beyond the scope of this paper.

The terms landmark and airborne navigation aid are used somewhat interchangeably throughout this report. Strictly speaking, a landmark is a SLAM term referring to a fixed map reference; whereas an airborne navigation aid serves a similar purpose in the current research, albeit its location is known but not fixed.

4.1 Approach to Overcome SLAM Constraints

Section 2.4 described several of the major constraints related to the SLAM algorithm including: landmark initialization, landmark association, and computational complexity. The information contained in the ADS-B data set for each airborne navigation aid allows many of these limiting factors and constraints to easily be mitigated. These mitigation strategies are described in the remainder of this section.

As detailed in Section 2.1, the ADS-B message set contains identification, position, velocity, altitude, and uncertainty information for each transmitting aircraft. The presence of this data eliminates concerns of landmark initialization because the landmark location is provided in the data. As such, the landmark can immediately be initialized upon receipt of the first complete set of ADS-B data. Likewise, landmark association is overcome because each ADS-B transmission includes unique aircraft identification information that can be used as an index for the airborne navigation aid.

The computational complexity of the SLAM algorithm is not as easily overcome as landmark initialization and landmark association. As discussed in Section 2.4, computational complexity is inherent in the SLAM algorithm and cannot be completely eliminated. In order to bound the complexity, airborne navigation aids that are no longer within the radio horizon of the host vehicle will be removed from the system state. According to research performed by Dissanayake, Durrant-Whyte, and Bailey, it is possible to remove a landmark from a SLAM map "without affecting the statistical consistency of the underlying estimation process" [47]. Therefore, this expansion and contraction of the state is considered a viable method to limit the computational complexity of the filter. Additional techniques could be considered to limit the number of airborne navigation aids contained in the system state to a tractable value. If the number of navigation aids were limited, then an algorithm could be developed to select the best airborne navigation aids for inclusion based on distance and geometry relative to the host vehicle. For purposes of this research, the number of navigation aids that the filter can track has not been limited, nor has any consideration been given to selecting the best possible navigation aids from those available.

4.2 Filter Formulation

This section provides the technical solution for determining the geodetic position of the host vehicle based on the reception of AOA and AOE information from ADS-B equipped aircraft. The geodetic position is determined through implementation of an EKF-SLAM-like algorithm. The EKF and SLAM algorithms were previously described in Section 2.3.1 and Section 2.4 respectively. The navigation algorithm tracks the position and velocity of airborne navigation aids through fusion of the reported ADS-B information, the AOA, and the AOE of the received signal. In addition to tracking airborne navigation aids, the algorithm simultaneously calculates the geodetic position, velocity, and acceleration of the host vehicle. The remainder of the present section describes the design assumptions then details the formulation and initialization of the various matrices and vectors used in the implementation.

4.2.1 Design Assumptions

A number of assumptions have previously been stated and several more are necessary to finalize the design. This section provides a convenient listing of the assumptions that have been incorporated into the design.

- All geodesic calculations are performed with respect to a spherical earth model.
- The processing rate of the algorithm is sufficient as to allow aircraft state transitions to be calculated using linear propagation techniques.
- The host vehicle state model assumes constant acceleration.
- The airborne navigation aid state model assumes constant velocity.
- ADS-B information can be received up to a maximum range of 190 NM provided that the transmitting and receiving aircraft have line-of-sight to one another.

- ADS-B AOA measurements are reported with respect to true north and have a RMSE of 0.7°.
- ADS-B AOE measurements are reported with respect to a locally level tangent plane and have a RMSE of 0.7°.
- An ADS-B aircraft's vertical velocity uncertainty is assumed to be 1.5 times the lateral velocity uncertainty reported by that aircraft.

4.2.2 State Vector

The state vector for the host vehicle, presented in (4-1), contains estimates for the host vehicle's position, velocity, and acceleration in the ECEF frame. The ECEF frame was selected because it does not contain the singularities present in geographic coordinate systems.

$$\hat{\boldsymbol{x}}_h = [\hat{\boldsymbol{x}}_{r_h} \quad \hat{\boldsymbol{x}}_{\dot{r}_h} \quad \hat{\boldsymbol{x}}_{\ddot{r}_h}] \tag{4-1}$$

Where: $\hat{\boldsymbol{x}}_h$ = Host vehicle state vector in ECEF.

$\hat{\boldsymbol{x}}_{r_h}$ = Host vehicle position estimate in ECEF.

$\hat{\boldsymbol{x}}_{\dot{r}_h}$ = Host vehicle velocity estimate in ECEF.

$\hat{\boldsymbol{x}}_{\ddot{r}_h}$ = Host vehicle acceleration estimate in ECEF.

The state vector for the navigation aids is somewhat more complex due to the characteristics of the errors present in the ADS-B reported position and velocity. These errors are predominately caused by a combination of latency and navigational errors inherent in the transmitting aircraft's navigation sensor suite, causing the ADS-B data to appear biased. Unfortunately, the mathematical model of these errors does not satisfy the fundamental Kalman Filter assumption that error sources are normally distributed, zero mean Gaussian random variables. Therefore, the state vector for each navigation aid is augmented using the Schmidt-Kalman techniques described in Section 2.3.2 to include states for these unobservable nuisance biases. The resulting state vector for each airborne navigation aid then contains estimates for the navigation aid's position, velocity, position measurement bias, and velocity measurement bias, all expressed in the ECEF frame. The state vector for an arbitrary airborne navigation aid is given in (4-2).

$$\hat{\boldsymbol{x}}_n = \begin{bmatrix} \hat{\boldsymbol{x}}_{r_n} & \hat{\boldsymbol{x}}_{\dot{r}_n} & \hat{\boldsymbol{\delta}}_{r_n} & \hat{\boldsymbol{\delta}}_{\dot{r}_n} \end{bmatrix} \qquad \textbf{(4-2)}$$

Where: $\hat{\boldsymbol{x}}_n$ = Navigation aid state vector in ECEF.
$\hat{\boldsymbol{x}}_{r_n}$ = Navigation aid position estimate in ECEF.
$\hat{\boldsymbol{x}}_{\dot{r}_n}$ = Navigation aid velocity estimate in ECEF.
$\hat{\boldsymbol{\delta}}_{r_n}$ = Navigation aid position measurement bias in ECEF. Set and held to 0.
$\hat{\boldsymbol{\delta}}_{\dot{r}_n}$ = Navigation aid velocity measurement bias in ECEF. Set and held to 0.

Finally, the host vehicle state and the state of each observed airborne navigation aid are expressed as the single combined state vector shown in (4-3); where m is the number of airborne navigation aids being tracked. It is evident that the state vector expands and contracts, as discussed in Section 4.1, to accommodate the changing number of navigation aids within the radio horizon. Following initialization, the elements of the state vector $\boldsymbol{x}$ are computed by the EKF.

$$\hat{\boldsymbol{x}} = [\hat{\boldsymbol{x}}_h \quad \hat{\boldsymbol{x}}_{n_1} \quad \cdots \quad \hat{\boldsymbol{x}}_{n_m}]^T \qquad \textbf{(4-3)}$$

4.2.3 State Covariance Matrix

The state covariance matrix $\boldsymbol{P}$, is a square matrix with dimensions (9+12m x 9+12m). The rows and columns of $\boldsymbol{P}$ express the covariance between the elements of the state vector $\boldsymbol{x}$. Following initialization, the elements of $\boldsymbol{P}$ are computed by the EKF.

4.2.4 Conditioning of ADS-B Data for use in the Filter

Many of the data elements received in the ADS-B message are not of the proper form for direct use in the filter. For example, ADS-B position is given in the geographic frame and ADS-B velocity is given in the ENU frame. These measurements are not directly compatible with the ECEF frame selected for the filter. Accordingly, the ADS-B data must be rotated into the ECEF frame prior to use in the filter. Additionally, ADS-B position and velocity uncertainty is expressed at the 95% containment level, while the filter requires normally distributed random variables. As such, the ADS-B uncertainty values must be appropriately scaled and expressed in the ECEF frame before being used in the filter. This section describes the preliminary ADS-B data conversions that take place prior to applying ADS-B data to the filter.

4.2.4.1 ADS-B Position Conversion

Received ADS-B position data is expressed in the geographical latitude/longitude coordinate system and must be converted to ECEF using the relationships presented in Section 0 prior to use by the filter. This collection of conversions is given in (4-4) through (4-6).

$$\tilde{r}_{n_x} = (R_m)\,cos(\tilde{\varphi}_n)\,cos(\tilde{\lambda}_n) \tag{4-4}$$

Where: $\tilde{r}_{n_x}$ = Measured navigation aid ECEF x coordinate.

R_m = Mean spherical radius of the Earth.

$\tilde{\varphi}_n$ = Reported navigation aid geodetic latitude.

$\tilde{\lambda}_n$ = Reported navigation aid geodetic longitude.

$$\tilde{r}_{n_y} = (R_m)\,cos(\tilde{\varphi}_n)\,sin(\tilde{\lambda}_n) \tag{4-5}$$

Where: $\tilde{r}_{n_y}$ = Measured navigation aid ECEF y coordinate.

$$\tilde{r}_{n_z} = (R_m)sin(\tilde{\varphi}_n) \tag{4-6}$$

Where: $\tilde{r}_{n_z}$ = Measured navigation aid ECEF z coordinate.

4.2.4.2 ADS-B Velocity Conversion

The ADS-B velocity data received from airborne navigation aids is presented in the ENU frame and must be expressed in the ECEF frame using the rotations given in Section 2.6.1.1.5. This is expressed in (4-7).

$$\tilde{\dot{\boldsymbol{r}}}_n = \begin{bmatrix} \tilde{\dot{r}}_{n_x} \\ \tilde{\dot{r}}_{n_y} \\ \tilde{\dot{r}}_{n_z} \end{bmatrix} = \boldsymbol{C}_{NED}^{ECEF} \begin{bmatrix} \tilde{\dot{r}}_{n_N} \\ \tilde{\dot{r}}_{n_E} \\ \tilde{\dot{r}}_{n_D} \end{bmatrix} \tag{4-7}$$

Where: $\tilde{\dot{\boldsymbol{r}}}_n$ = Measured navigation aid ECEF velocity vector.

$\boldsymbol{C}_{NED}^{ECEF}$ = Coordinate conversion matrix from NED to ECEF = ${\boldsymbol{C}_{ECEF}^{NED}}^T$. Refer to (2-77).

$\tilde{\dot{r}}_{n_N}$ = Reported navigation aid velocity north.

$\tilde{\dot{r}}_{n_E}$ = Reported navigation aid velocity east.

$\tilde{\dot{r}}_{n_D}$ = Reported navigation aid velocity down = $-\tilde{\dot{r}}_{n_U}$

4.2.4.3 <u>ADS-B Position Uncertainty Conversions</u>

The Aircraft Operational Status Message provides an indication of the airborne navigation aid's lateral and vertical position uncertainty. The lateral position uncertainty is given as a zero-mean, 95% Rayleigh distributed random variable, while the vertical position uncertainty expressed as a zero-mean, 95% Gaussian random variable. Prior to use in the filter, these 95% uncertainty level values must be converted to their respective normally distributed 68% uncertainty levels. As a safety factor to mitigate against the possibility of the filter generating overly optimistic estimates from these values, the position inputs will be treated though they are provided at the 92.5% containment level rather than the defined 95% containment level. Using the constants described in Section 2.5, these conversions are given as (4-8) and (4-9) respectively.

$$\tilde{\sigma}_{r_{n_l}}^{N68} = \frac{\tilde{\sigma}_{r_{n_l}}^{R95}}{2.276079} \tag{4-8}$$

Where: $\tilde{\sigma}_{r_{n_l}}^{N68}$ = Measured navigation aid lateral position uncertainty (68% containment).

$\tilde{\sigma}_{r_{n_l}}^{R95}$ = Reported navigation aid lateral position uncertainty (95% containment).

2.276079 = Statistical conversion from Rayleigh 92.5% to Normal 68%.

$$\tilde{\sigma}_{r_{n_v}}^{N68} = \frac{\tilde{\sigma}_{r_{n_v}}^{N95}}{1.780464} \qquad \textbf{(4-9)}$$

Where: $\tilde{\sigma}_{r_{n_v}}^{N68}$ = Measured navigation aid vertical position uncertainty (68% containment).

$\tilde{\sigma}_{r_{n_v}}^{N95}$ = Reported navigation aid vertical position uncertainty (95% containment).

1.780464 = Statistical conversion from Normal 92.5% to Normal 68%.

Once expressed at the 68% uncertainty level, the navigation aid lateral and vertical uncertainties are gathered into a NED vector, and then rotated into the ECEF frame using the techniques presented in Section 2.6.1.1.5. This rotation is presented as (4-10).

$$\tilde{\boldsymbol{R}}_{r_n} = \left(\boldsymbol{C}_{NED}^{ECEF}\right)\left(diag\begin{bmatrix} \left(\tilde{\sigma}_{r_{n_l}}^{N68}\right)^2 \\ \left(\tilde{\sigma}_{r_{n_l}}^{N68}\right)^2 \\ \left(\tilde{\sigma}_{r_{n_v}}^{N68}\right)^2 \end{bmatrix}\right)\left(\boldsymbol{C}_{NED}^{ECEF}\right)^T \qquad \textbf{(4-10)}$$

Where: $\tilde{\boldsymbol{R}}_{r_n}$ = Measured navigation aid ECEF position measurement covariance matrix.

$\boldsymbol{C}_{NED}^{ECEF}$ = Coordinate conversion matrix from NED to ECEF.

$diag$ = Operator used to construct a diagonal matrix of the input values.

Following rotation, the navigation aid uncertainties are contained in a fully populated 3x3 matrix with covariance present between the three ECEF axes. The main diagonal elements of the rotated matrix provide the variance for each ECEF axis; however, unless otherwise specified, the full covariance matrix is retained and used in filter calculations.

4.2.4.4 ADS-B Velocity Uncertainty Conversions

Following a similar process to that used for ADS-B position uncertainty, the velocity uncertainty must be scaled and rotated prior to use in the filter. The lateral velocity uncertainty is given in the Airborne Velocity Message as a zero-mean, 95% Rayleigh random variable. The ADS-B data set does not provide a vertical velocity uncertainty; therefore, the vertical velocity uncertainty is assumed to be 1.5 times the lateral velocity uncertainty. Although not directly stated, this relationship is implied on page 443 of DO-260B where it is stated that a NACv = 2 requires a horizontal velocity error of less than 3 m/s and a vertical velocity error of less than 15 ft/sec [1]. The 1.5 constant of proportionality is evident when one accounts for the mixed units in this expression. The statistical scaling and subsequent rotation to the ECEF frame are provided in (4-11) through (4-13).

$$\tilde{\sigma}_{\dot{r}_{n_l}}^{N68} = \frac{\tilde{\sigma}_{\dot{r}_{n_l}}^{R95}}{2.447747} \tag{4-11}$$

Where: $\tilde{\sigma}_{\dot{r}_{n_l}}^{N68}$ = Measured navigation aid lateral velocity uncertainty (68% containment).

$\tilde{\sigma}_{\dot{r}_{n_l}}^{R95}$ = Reported navigation aid lateral velocity uncertainty (95% containment).

2.447747 = Statistical conversion from Rayleigh 95% to a Normal 68% random variable.

$$\tilde{\sigma}_{\dot{r}_{n_v}}^{N68} = 1.5\tilde{\sigma}_{\dot{r}_{n_l}}^{N68} \tag{4-12}$$

Where: $\tilde{\sigma}_{\dot{r}_{n_v}}^{N68}$ = Measured navigation aid vertical velocity uncertainty (68% containment).

1.5 = Constant of proportionality between lateral and vertical velocity uncertainty.

$$\widetilde{\boldsymbol{R}}_{\dot{r}_n} = \left(\boldsymbol{C}_{NED}^{ECEF}\right)\left(diag\begin{bmatrix} \left(\tilde{\sigma}_{\dot{r}_{n_l}}^{N68}\right)^2 \\ \left(\tilde{\sigma}_{\dot{r}_{n_l}}^{N68}\right)^2 \\ \left(\tilde{\sigma}_{\dot{r}_{n_v}}^{N68}\right)^2 \end{bmatrix}\right)\left(\boldsymbol{C}_{NED}^{ECEF}\right)^T \tag{4-13}$$

Where: $\widetilde{\boldsymbol{R}}_{\dot{r}_n}$ = Measured navigation aid ECEF velocity measurement covariance matrix.

$\boldsymbol{C}_{NED}^{ECEF}$ = Coordinate conversion matrix from NED to ECEF.

$diag$ = Operator used to construct a diagonal matrix of the input values.

4.2.5 Time Propagation

Fundamental to the concept of Kalman filtering is the process of developing an estimate of the system state based on the system model. For this application, a time propagation rate of 10 Hz has been selected. This rate allows a linear state transition to be assumed because the non-linearity due to the curvature of the Earth over the distance traveled in 100 milliseconds at subsonic speeds can be considered negligible. Given the assumption of local linearity, the *a* *priori* estimate of the system state and covariance are determined using the linear Kalman Filter time update equations previously given in (2-10) and (2-11), and restated as (4-14) and (4-15) for convenience.

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{A}\hat{\boldsymbol{x}}_{k-1} \tag{4-14}$$

Where: $\hat{\boldsymbol{x}}_k$ = System state at time *k*, described in Section 4.2.1.

$\boldsymbol{A}$ = State transition matrix described in Section 4.2.5.1.

$$\boldsymbol{P}_k^- = \boldsymbol{A}\boldsymbol{P}_{k-1}\boldsymbol{A}^T + \boldsymbol{Q} \tag{4-15}$$

Where: $\boldsymbol{P}_k$ = Covariance matrix at time *k*, described in Section 4.2.3.

$\boldsymbol{Q}$ = Process noise matrix described in Section 4.2.5.2.

4.2.5.1 State Transition Matrix

The state transition matrix provides a system model by which to propagate the system state and covariance forward in time to generate the *a priori* estimates. Because linear propagation has been assumed for the motion model, the state transition matrix primarily models the linear kinematic equations of motion presented in (4-16) and (4-17). The expression in (4-16) assumes constant acceleration; while the alternate form in (4-17) assumes constant velocity. For purposes of this implementation, constant acceleration is assumed for the host vehicle, while constant velocity is assumed for the motion of the airborne navigation aids.

$$\boldsymbol{r} = \boldsymbol{r}_0 + \dot{\boldsymbol{r}}t + \frac{1}{2}\ddot{\boldsymbol{r}}t^2 \tag{4-16}$$

Where: $\boldsymbol{r}$ = Position vector in ECEF.

$\boldsymbol{r}_0$ = Previous/initial position vector in ECEF.

$\dot{\boldsymbol{r}}$ = Velocity vector in ECEF.

$\ddot{\boldsymbol{r}}$ = Acceleration vector in ECEF.

t = Time interval.

$$\boldsymbol{r} = \boldsymbol{r}_0 + \dot{\boldsymbol{r}}t \tag{4-17}$$

In addition to modelling the equations of linear motion, the state transition matrix also provides a model for the unobservable navigation aid bias states. These bias states are modeled as first order Gauss-Markov random stochastic processes. A Gauss-Markov process can be generated by passing uncorrelated white noise through a linear first-order feedback filter [48]. For these random processes, the probability distribution is entirely dependent on the preceding estimate [48]. The differential equation governing these continuous systems is provided in (4-18) [48]; while the corresponding difference equation for discrete systems is given in (4-19) [48]. The state transition matrix models the exponential term of (4-19), while the process noise matrix described the following section accounts for the white noise component.

$$\dot{v} = -\beta v + w \tag{4-18}$$

Where: v = Random Gauss-Markov variable
$\dot{v}$ = Derivative of random Gauss-Markov variable.
β = Correlation time = $1/\tau$
w = White noise.

$$v_{k+1} = e^{-\beta t} v_k + w_k \tag{4-19}$$

Where: v_k = Random Gauss-Markov variable at step k.
β = Correlation time = $1/\tau$
t = Time between steps.
w_k = White noise at step k.

An expression for the state transition matrix for a single dimension of host vehicle motion is given in (4-20). Similarly, the single dimension state transition expression for airborne navigation aid motion is provided in the upper 2x2 submatrix of (4-21). Finally, the lower 2x2 submatrix of (4-21) provides the state transition for the bias states. These single dimension expressions are easily extended to three dimensions as shown in the resulting (9+12m x 9+12m) state transition matrix presented in (4-22).

Although the elements of the state transition matrix are constant, the matrix is constructed at each time step to account for changes in the number of tracked navigation aids. As an implementation optimization, this matrix could be preserved in persistent memory and updated only when a change in the number of tracked navigation aids is detected. For purposes of this research, the matrix is reconstructed at each time step.

$$\boldsymbol{A}_{h_{x,y,z}} = \begin{bmatrix} 1 & dt & 0.5dt^2 \\ 0 & 1 & dt \\ 0 & 0 & 1 \end{bmatrix} \tag{4-20}$$

Where: $\boldsymbol{A}_h$ = State transition matrix for host vehicle motion.
dt = Propagation time interval = 0.1 seconds.

$$\boldsymbol{A}_{n_{x,y,z}} = \begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{-\frac{dt}{\tau_r}} & 0 \\ 0 & 0 & 0 & e^{-\frac{dt}{\tau_{\dot{r}}}} \end{bmatrix} \tag{4-21}$$

Where: $\boldsymbol{A}_n$ = State transition matrix for navigation aid vehicle motion.
τ_r = Time constant for position biases = 300 seconds (empirically derived).
$\tau_{\dot{r}}$ = Time constant for velocity biases = 90 seconds (empirically derived).

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_h & 0 & 0 & 0 \\ 0 & \boldsymbol{A}_{n_1} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \boldsymbol{A}_{n_m} \end{bmatrix} \tag{4-22}$$

Where: $\boldsymbol{A}$ = Combined state transition matrix.

$\boldsymbol{A}_h$ = State transition matrix for the host vehicle motion.

$$= \begin{bmatrix} 1 & 0 & 0 & dt & 0 & 0 & 0.5dt^2 & 0 & 0 \\ 0 & 1 & 0 & 0 & dt & 0 & 0 & 0.5dt^2 & 0 \\ 0 & 0 & 1 & 0 & 0 & dt & 0 & 0 & 0.5dt^2 \\ 0 & 0 & 0 & 1 & 0 & 0 & dt & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & dt & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & dt \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$\boldsymbol{A}_{n_m}$= State transition matrix for navigation aid *m* motion.

$$= \begin{bmatrix} 1 & 0 & 0 & dt & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & dt & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & dt & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{-\frac{dt}{\tau_r}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{-\frac{dt}{\tau_r}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{-\frac{dt}{\tau_r}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{-\frac{dt}{\tau_{\dot{r}}}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{-\frac{dt}{\tau_{\dot{r}}}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{-\frac{dt}{\tau_{\dot{r}}}} \end{bmatrix}$$

4.2.5.2 Process Noise

The process noise matrix $\boldsymbol{Q}$, is used to account for imperfections in the system model that would be difficult or impossible to include in the model. This term can be considered synonymous with plant noise in control theory. One classic example of process noise applicable to this solution is that kinematic systems are continuous, implying that the outputs can vary at any point in time; however, the Kalman filter is a discrete system that assumes constant inputs at each time step [49]. The process noise matrix is used to add uncertainty to the Kalman system to account for these types of simplifying assumptions. Other sources of process noise present in this solution are the assumption of linear motion, the use of a spherical Earth model, and the batch application of measurements that are received between periodic time propagation intervals.

4.2.5.2.1 *Host Vehicle Process Noise*

The process noise matrix for the host vehicle is constructed using the piecewise white noise model described in [49]. In this model, the highest order state transition term is assumed constant for the duration of each time step [49]. Following this assumption, acceleration is assumed constant for each time step, implying that the one-dimensional change in velocity and position are as shown in (4-23) and (4-24); where these values were determined through integration of the acceleration.

$$\Delta\dot{r} = \ddot{r}dt \tag{4-23}$$

Where: $\Delta\dot{r}$ = Change in velocity.
$\ddot{r}$ = Acceleration.
dt = Time interval.

$$\Delta r = 0.5\ddot{r}dt^2 \tag{4-24}$$

Where: Δr = Change in position.

The change in system state assuming constant acceleration can then be modeled in matrix form as given in (4-25) [49].

$$\boldsymbol{\Gamma} = \begin{bmatrix} \Delta r \\ \Delta\dot{r} \\ \Delta\ddot{r} \end{bmatrix} = \begin{bmatrix} 0.5dt^2 \\ dt \\ 1 \end{bmatrix} \ddot{r} \tag{4-25}$$

Where: $\boldsymbol{\Gamma}$ = Change in system state.

The covariance of the process noise is then expressed as the statistical expectation shown in (4-26) [49].

$$\boldsymbol{Q} = E[\boldsymbol{\Gamma}\ddot{r}\ddot{r}\boldsymbol{\Gamma}^T] = \boldsymbol{\Gamma}\left(\sigma_{\ddot{r}_h}^{N68}\right)^2\boldsymbol{\Gamma}^T \tag{4-26}$$

Where: $\left(\sigma_{\ddot{r}_h}^{N68}\right)^2$ = Variance of host vehicle acceleration.

Performing the matrix multiplication of (4-26), and factoring out the variance, results in the one-dimensional process noise for the host vehicle shown in (4-27). Extending the one-dimensional process noise to three dimensions leads directly to (4-28).

$$\boldsymbol{Q}_{h_{x,y,z}} = \begin{bmatrix} \frac{dt^4}{4} & \frac{dt^3}{2} & \frac{dt^2}{2} \\ \frac{dt^3}{2} & dt^2 & dt \\ \frac{dt^2}{2} & dt & 1 \end{bmatrix} \left(\sigma_{\ddot{r}_h}^{N68}\right)^2 \tag{4-27}$$

Where: $\boldsymbol{Q}_{h_{x,y,z}}$ = One dimensional host vehicle process noise.

$$\boldsymbol{Q}_h = \begin{bmatrix} \frac{dt^4}{4} & 0 & 0 & \frac{dt^3}{2} & 0 & 0 & \frac{dt^2}{2} & 0 & 0 \\ 0 & \frac{dt^4}{4} & 0 & 0 & \frac{dt^3}{2} & 0 & 0 & \frac{dt^2}{2} & 0 \\ 0 & 0 & \frac{dt^4}{4} & 0 & 0 & \frac{dt^3}{2} & 0 & 0 & \frac{dt^2}{2} \\ \frac{dt^3}{2} & 0 & 0 & dt^2 & 0 & 0 & dt & 0 & 0 \\ 0 & \frac{dt^3}{2} & 0 & 0 & dt^2 & 0 & 0 & dt & 0 \\ 0 & 0 & \frac{dt^3}{2} & 0 & 0 & dt^2 & 0 & 0 & dt \\ \frac{dt^2}{2} & 0 & 0 & dt & 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{dt^2}{2} & 0 & 0 & dt & 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{dt^2}{2} & 0 & 0 & dt & 0 & 0 & 1 \end{bmatrix} \left(\sigma_{\ddot{r}_h}^{N68}\right)^2 \tag{4-28}$$

Where: $\boldsymbol{Q}_h$ = Host vehicle process noise covariance matrix.

$\left(\sigma_{\ddot{r}_h}^{N68}\right)^2$= Empirically derived variance of host vehicle acceleration = $(0.075)^2$

4.2.5.2.2 *Airborne Navigation Aid Process Noise*

The process noise matrix for the airborne navigation aids is a function of the navigation aid position and velocity uncertainty values received in the ADS-B message set. In addition to position and velocity uncertainty, the matrix models the variance of the of the Gauss-Markov position and velocity bias states. The general form of the Gauss-Markov variance shown in (4-29) was provided by Gelb in [48]. An informal justification of this value is provided in APPENDIX G.

$$(\sigma_w)^2 = 2\beta(\sigma_v)^2 \qquad \textbf{(4-29)}$$

Where: $(\sigma_w)^2$ = Gauss-Markov process variance.
β = Correlation time = $1/\tau$
$(\sigma_v)^2$ = Gauss-Markov random variable variance.

Depending on the bias state being modeled, the Gauss-Markov random variable variance term of (4-29) represents either a position or velocity bias variance. The position variance described in Section 4.2.4.3 accounts for potential biases present in the navigation system of the transmitting aircraft, but it cannot not account for ADS-B transmission latency. Because transport latency can contribute significantly to the overall navigation aid position uncertainty – especially for highly accurate navigation aids – the position variance reported by the navigation aid is augmented with a position latency variance when modeling the bias state covariance. These position latency variances are based on the aircraft velocity and the median of the expected latency. The final 3x3 position latency covariance matrix expressed in the ECEF frame is given in (4-30).

$$\widehat{\boldsymbol{Q}}_{\Delta t_r} = \left(\boldsymbol{C}_{NED}^{ECEF}\right)\left(diag\begin{bmatrix} \left(\hat{\sigma}_{\Delta t_{r_{NE}}}^{N68}\right)^2 \\ \left(\hat{\sigma}_{\Delta t_{r_{NE}}}^{N68}\right)^2 \\ \left(\hat{\sigma}_{\Delta t_{r_D}}^{N68}\right)^2 \end{bmatrix}\right)\left(\boldsymbol{C}_{NED}^{ECEF}\right)^T \tag{4-30}$$

Where: $\widehat{\boldsymbol{Q}}_{\Delta t_r}$ = Navigation aid process noise position latency covariance matrix (3x3).

$\boldsymbol{C}_{NED}^{ECEF}$ = Coordinate conversion matrix from NED to ECEF.

$\left(\hat{\sigma}_{\Delta t_{r_{NE}}}^{N68}\right)^2$ = Estimated North/East position latency variance:

$= (\Delta t_r)^2 \left[\left(\tilde{\dot{r}}_{n_N}\right)^2 + \left(\tilde{\dot{r}}_{n_E}\right)^2\right]$

$\left(\hat{\sigma}_{\Delta t_{r_D}}^{N68}\right)^2$ = Estimated vertical position latency variance = $(\Delta t_r)^2\left(-\tilde{\dot{r}}_{n_U}\right)^2$

Δt_r = 50th percentile position measurement latency:

50th percentile ADS-B transmit latency	0.20 seconds
50th percentile navigation processing latency	0.05 seconds
Total	0.25 seconds

$\tilde{\dot{r}}_{n_N}$ = ADS-B reported navigation aid velocity north.

$\tilde{\dot{r}}_{n_E}$ = ADS-B reported navigation aid velocity east.

$\tilde{\dot{r}}_{n_U}$ = ADS-B reported navigation aid velocity up.

Substituting the navigation aid position measurement variance described in Section 4.2.4.3, and its corresponding position latency variance given in (4-30), into (4-29) leads to the expression for the (3x3) navigation aid position bias process noise covariance matrix given as (4-31).

$$\widehat{\boldsymbol{Q}}_{\delta_{r_n}} = 2\beta_r\left[\widetilde{\boldsymbol{R}}_{r_n} + \widehat{\boldsymbol{Q}}_{\Delta t_r}\right] \tag{4-31}$$

Where: $\widehat{\boldsymbol{Q}}_{\delta_{r_n}}$ = Navigation aid position measurement bias process noise covariance matrix.

β_r = Position correlation time = $1/\tau_r$ = 1/300; where 300 was empirically derived.

$\widetilde{\boldsymbol{R}}_{r_n}$ = Navigation aid ECEF position covariance matrix. Refer to Section 4.2.4.3.

The navigation aid velocity bias process noise is constructed by substituting the velocity variance described in Section 4.2.4.4 into (4-29). This is shown in (4-32).

$$\widehat{\boldsymbol{Q}}_{\delta_{\dot{r}_n}} = 2\beta_{\dot{r}}\widetilde{\boldsymbol{R}}_{\dot{r}_n} \tag{4-32}$$

Where: $\widehat{\boldsymbol{Q}}_{\delta_{\dot{r}_n}}$ = Navigation aid velocity measurement bias process noise covariance matrix.

$\beta_{\dot{r}}$ = Velocity correlation time = $1/\tau_{\dot{r}}$ = 1/90; where 90 was empirically derived.

$\widetilde{\boldsymbol{R}}_{\dot{r}_n}$ = Navigation aid ECEF velocity covariance matrix. Refer to Section 4.2.4.4.

The ensuing (12x12) process noise for an arbitrary navigation aid is given in (4-33).

$$\boldsymbol{Q}_n = \begin{bmatrix} 0.00875\,\widetilde{\boldsymbol{R}}_{r_n} & 0 & 0 & 0 \\ 0 & 0.05\,\widetilde{\boldsymbol{R}}_{\dot{r}_n} & 0 & 0 \\ 0 & 0 & \widehat{\boldsymbol{Q}}_{\delta_{r_n}} & 0 \\ 0 & 0 & 0 & \widehat{\boldsymbol{Q}}_{\delta_{\dot{r}_n}} \end{bmatrix} dt \tag{4-33}$$

Where: $\boldsymbol{Q}_n$ = Navigation aid process noise matrix.

0.00875 = Empirically derived position uncertainty scaling factor.

0.05 = Empirically derived velocity uncertainty scaling factor.

4.2.5.2.3 Combined Process Noise Matrix

The process noise matrix for the host vehicle and for each airborne navigation aid are then combined into a single (9+12m x 9+12m) matrix shown in (4-34). The combined process noise matrix must be constructed at each time step to account for changing numbers of navigation aids and changes in navigation aid uncertainty.

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{Q}_h & 0 & 0 & 0 \\ 0 & \boldsymbol{Q}_{n_1} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \boldsymbol{Q}_{n_m} \end{bmatrix} \tag{4-34}$$

Where: $\boldsymbol{Q}$ = Process noise matrix (9+12*m x* 9+12*m*).

$\boldsymbol{Q}_h$ = Host vehicle process noise sub-matrix (4-28).

$\boldsymbol{Q}_{n_m}$ = Airborne navigation aid *m* process noise sub-matrix (4-33).

4.2.6 Observation Update

Observation updates are also applied using a 10 Hz frame rate. An observation (measurement) update is applied independently (not batched) for each navigation aid that has provided fresh ADS-B data in the preceding 100 milliseconds. ADS-B position and velocity messages are used to update the corresponding navigation aid state, while AOA and AOE measurements are simultaneously applied to both the host vehicle and the source navigation aid. Observations are applied using the following order of precedence: position messages, velocity messages, and then operational status messages. That is, if both a velocity message and a position message have been received from a given navigation aid in the preceding 100 milliseconds, then only the position message is used to update the filter. The remainder of this section defines the contents of the various matrices that are used in the observation update process previously given in (2-28) through (2-31).

4.2.6.1 Measurement Vector

The measurement vector is a column vector of the measured data that is to be applied to the filter. Since it contains measured data, its contents vary as a function of the type of ADS-B packet being processed. The AOA and AOE data is applied independently of the position or velocity data; therefore, the AOA/AOE measurements are expressed in a separate measurement vector. This separation of measurements is not strictly necessary, but is preferred because position and velocity measurements are applied only to the airborne navigation aids being tracked, while AOA and AOE measurements are applied to both the navigation aids and the host vehicle. Equation (4-35) gives the measurement vector for an ADS-B position message, while (4-36) is the measurement vector for an ADS-B velocity message. Finally, (4-37) provides the measurement vector for AOA/AOE measurements.

$$\tilde{\boldsymbol{z}}_{pos} = \begin{bmatrix} \tilde{r}_{n_x} & \tilde{r}_{n_y} & \tilde{r}_{n_z} \end{bmatrix}^T \qquad \textbf{(4-35)}$$

Where: $\tilde{\boldsymbol{z}}_{pos}$ = Airborne navigation aid position measurement vector.

$\tilde{r}_{n_{x,y,z}}$= Navigation aid ADS-B reported position in the ECEF frame (Section 4.2.4.1).

$$\tilde{\boldsymbol{z}}_{vel} = \begin{bmatrix}\tilde{\dot{r}}_{n_x} & \tilde{\dot{r}}_{n_y} & \tilde{\dot{r}}_{n_z}\end{bmatrix}^T$$ **(4-36)**

Where: $\tilde{\boldsymbol{z}}_{vel}$ = Airborne navigation aid velocity measurement vector.

$\tilde{\dot{r}}_{n_{x,y,z}}$= Navigation aid ADS-B reported velocity in the ECEF frame (Section 4.2.4.2).

$$\tilde{\boldsymbol{z}}_{\alpha\beta} = [\tilde{\alpha}_n \quad \tilde{\beta}_n\,]^T$$ **(4-37)**

Where: $\tilde{\boldsymbol{z}}_{\alpha\beta}$ = Airborne navigation aid AOA/AOE measurement vector.

$\tilde{\alpha}_n$ = True North referenced AOA of navigation aid *m* ADS-B data.

$\tilde{\beta}_n$ = Locally level referenced AOE of navigation aid *m* ADS-B data.

The AOA is assumed to be expressed with respect to True North and the AOE is assumed to be expressed with respect to a locally level tangent plane. Expressing the AOA and AOE in these frames is assumed to be a function of the virtual ADS-B receiver as described in Section 1.3; accordingly, no data conversions are required for AOA/AOE data.

The measurement vector must be evaluated at each time step and for each airborne navigation aid for which data has been received since the last measurement update.

4.2.6.2 Measurement (Geometry) Matrix

The measurement matrix relates the estimated state of the system to the measured data. In less formal terms, the measurement matrix is used by the EKF-SLAM algorithm to generate an estimate of the expected measurement values based only on the current system state. As previously described in Section 2.3, these estimated measurements are then compared with the actual measurements to generate the Kalman.

Position and velocity information for each airborne navigation aid is included in the state vector, allowing their values to be estimated directly from the state. On the other hand, calculating AOA and AOE estimates from the state vector requires the implementation of the non-linear functions provided by Leick et al. in (4-38) and (4-39) [41]. Derivations of these equations are provided in APPENDIX H.

$$\hat{\alpha} = tan^{-1}\left[\frac{-\,sin(\lambda_o)\,\Delta x + cos(\lambda_o)\Delta y}{-\,sin(\varphi_o)\,cos(\lambda_o)\,\Delta x - sin(\varphi_o)\,sin(\lambda_o)\,\Delta y + cos(\varphi_o)\Delta z}\right] \tag{4-38}$$

Where: $\hat{\alpha}$ = Angle of arrival estimate.

φ_o = Latitude of the origin of the local coordinate frame (host vehicle).

λ_o = Longitude of the origin of the local coordinate frame (host vehicle).

$$\Delta \boldsymbol{X} = \begin{bmatrix}\Delta x\\ \Delta y\\ \Delta z\end{bmatrix} = \begin{bmatrix}\hat{r}_{n_x} - \hat{r}_{h_x}\\ \hat{r}_{n_y} - \hat{r}_{h_y}\\ \hat{r}_{n_z} - \hat{r}_{h_z}\end{bmatrix}$$

$\hat{r}_{n_{x,y,z}}$= ECEF position estimate of the navigation aid used for updating.

$\hat{r}_{h_{x,y,z}}$= ECEF position estimate of the host vehicle.

$$\hat{\beta} = sin^{-1}\left[\frac{cos(\varphi_o)\,cos(\lambda_o)\Delta x + cos(\varphi_o)\,sin(\lambda_o)\,\Delta y + sin(\varphi_o)\Delta z}{s}\right] \tag{4-39}$$

Where: $\hat{\beta}$ = Angle of elevation estimate.

s = Magnitude of the vector from the host vehicle to the navigation aid.

As described in the Extended Kalman Filter overview of Section 2.3.1, and shown in (2-23), the vector function $\boldsymbol{h}(\hat{\boldsymbol{x}})$ is a non-linear vector function that is used to generate the estimate of the measured values. The formulation of the vector function $\boldsymbol{h}$ varies based on the type of ADS-B data being processed. As such, (4-40) defines $\boldsymbol{h}$ for a position measurement, (4-41) describes $\boldsymbol{h}$ for a velocity measurement, and (4-42) gives $\boldsymbol{h}$ for the AOA/AOE measurements.

$$\boldsymbol{h}_{pos} = \hat{\boldsymbol{z}}_{pos} = \begin{bmatrix}\hat{r}_{n_x} & \hat{r}_{n_y} & \hat{r}_{n_z}\end{bmatrix}^T = \left[\left(\hat{\boldsymbol{x}}_{r_{n_x}}\right) \quad \left(\hat{\boldsymbol{x}}_{r_{n_y}}\right) \quad \left(\hat{\boldsymbol{x}}_{r_{n_z}}\right)\right]^T \tag{4-40}$$

Where: $\hat{\boldsymbol{z}}_{pos}$ = Estimate of airborne navigation aid position measurement data.

$\hat{r}_{n_{x,y,z}}$ = Estimate of airborne navigation aid position in ECEF.

$\hat{\boldsymbol{x}}_{r_{n_{x,y,z}}}$ = State vector estimate of the airborne navigation aid position in ECEF.

$$\boldsymbol{h}_{vel} = \hat{\boldsymbol{z}}_{vel} = \begin{bmatrix} \hat{\dot{r}}_{n_x} & \hat{\dot{r}}_{n_y} & \hat{\dot{r}}_{n_z} \end{bmatrix}^T = \begin{bmatrix} \left(\widehat{\boldsymbol{x}}_{\dot{r}_{n_x}}\right) & \left(\widehat{\boldsymbol{x}}_{\dot{r}_{n_y}}\right) & \left(\widehat{\boldsymbol{x}}_{\dot{r}_{n_z}}\right) \end{bmatrix}^T \tag{4-41}$$

Where: $\hat{\boldsymbol{z}}_{pos}$ = Estimate of airborne navigation aid velocity measurement data.

$\hat{\dot{r}}_{n_{x,y,z}}$ = Estimate of airborne navigation aid velocity in ECEF.

$\widehat{\boldsymbol{x}}_{r_{n_{x,y,z}}}$ = State vector estimate of the airborne navigation aid velocity in ECEF.

$$\boldsymbol{h}_{\alpha\beta} = \hat{z}_{\alpha\beta} = [\hat{\alpha} \quad \hat{\beta}]^T \tag{4-42}$$

Where: $\hat{z}_{\alpha\beta}$= Estimate of AOA & AOE measurement data.

$\hat{\alpha}$ = AOA estimate generated from the state vector. Refer to (4-38).

$\hat{\beta}$ = AOE estimate generated from the state vector. Refer to (4-39).

The Extended Kalman Filter overview of Section 2.3.1 explains that the measurement matrix $\boldsymbol{H}$ is computed as the Jacobian of the vector function $\boldsymbol{h}$. For position and velocity measurements, this results in a (3 x 9+12*m*) matrix with the elements shown in (4-43). The AOA and AOE measurements require a (2 x 9+12*m*) matrix of the same form.

$$\boldsymbol{H}_{[i,j]} = \frac{\partial \boldsymbol{h}_{[i]}(\boldsymbol{x})}{\partial \boldsymbol{x}_{[j]}} \tag{4-43}$$

$$\Rightarrow \boldsymbol{H}_{[i,j]} = \begin{bmatrix} \frac{\partial \boldsymbol{h}_{[1]}}{\partial \widehat{\boldsymbol{x}}_{[1]}} & \frac{\partial \boldsymbol{h}_{[1]}}{\partial \widehat{\boldsymbol{x}}_{[2]}} & \cdots & \frac{\partial \boldsymbol{h}_{[1]}}{\partial \widehat{\boldsymbol{x}}_{[n]}} \\ \frac{\partial \boldsymbol{h}_{[2]}}{\partial \widehat{\boldsymbol{x}}_{[1]}} & \frac{\partial \boldsymbol{h}_{[2]}}{\partial \widehat{\boldsymbol{x}}_{[2]}} & \cdots & \frac{\partial \boldsymbol{h}_{[2]}}{\partial \widehat{\boldsymbol{x}}_{[n]}} \\ \frac{\partial \boldsymbol{h}_{[3]}}{\partial \widehat{\boldsymbol{x}}_{[1]}} & \frac{\partial \boldsymbol{h}_{[3]}}{\partial \widehat{\boldsymbol{x}}_{[2]}} & \cdots & \frac{\partial \boldsymbol{h}_{[3]}}{\partial \widehat{\boldsymbol{x}}_{[n]}} \end{bmatrix}$$

The partial derivatives for the non-zero elements of $\boldsymbol{H}$ when an ADS-B position measurement is being applied are given in (4-44). Similarly, the non-zero elements for an ADS-B velocity measurement are provided in (4-45). Finally, the partial derivatives for $\hat{\alpha}$ and $\hat{\beta}$ (AOA and AOE measurements) given in (4-46) were provided by Leick et al. in [41].

$$\begin{gathered}
\frac{\partial \boldsymbol{h}_{pos_{[1]}}}{\partial \hat{\boldsymbol{x}}_{n_{[1]}}} = -\frac{\partial \boldsymbol{h}_{pos_{[1]}}}{\partial \hat{\boldsymbol{x}}_{n_{[7]}}} = 1 \\
\frac{\partial \boldsymbol{h}_{pos_{[2]}}}{\partial \hat{\boldsymbol{x}}_{n_{[2]}}} = -\frac{\partial \boldsymbol{h}_{pos_{[2]}}}{\partial \hat{\boldsymbol{x}}_{n_{[8]}}} = 1 \\
\frac{\partial \boldsymbol{h}_{pos_{[3]}}}{\partial \hat{\boldsymbol{x}}_{n_{[3]}}} = -\frac{\partial \boldsymbol{h}_{pos_{[3]}}}{\partial \hat{\boldsymbol{x}}_{n_{[9]}}} = 1
\end{gathered} \tag{4-44}$$

$$\begin{gathered}
\frac{\partial \boldsymbol{h}_{vel_{[1]}}}{\partial \hat{\boldsymbol{x}}_{n_{[4]}}} = -\frac{\partial \boldsymbol{h}_{vel_{[1]}}}{\partial \hat{\boldsymbol{x}}_{n_{[10]}}} = 1 \\
\frac{\partial \boldsymbol{h}_{vel_{[2]}}}{\partial \hat{\boldsymbol{x}}_{n_{[5]}}} = -\frac{\partial \boldsymbol{h}_{vel_{[2]}}}{\partial \hat{\boldsymbol{x}}_{n_{[11]}}} = 1 \\
\frac{\partial \boldsymbol{h}_{vel_{[3]}}}{\partial \hat{\boldsymbol{x}}_{n_{[6]}}} = -\frac{\partial \boldsymbol{h}_{vel_{[3]}}}{\partial \hat{\boldsymbol{x}}_{n_{[12]}}} = 1
\end{gathered} \tag{4-45}$$

$$\begin{aligned}
\frac{\partial \boldsymbol{h}_{\alpha\beta_{[1]}}}{\partial \hat{\boldsymbol{x}}_{h_{[1]}}} &= -\frac{\partial \boldsymbol{h}_{\alpha\beta_{[1]}}}{\partial \hat{\boldsymbol{x}}_{n_{[1]}}} = \frac{-\,sin(\varphi_o)\,cos(\lambda_o)\,sin(\hat{\alpha}) + sin(\lambda_o)cos(\hat{\alpha})}{s\big[cos(\hat{\beta})\big]} \\
\frac{\partial \boldsymbol{h}_{\alpha\beta_{[1]}}}{\partial \hat{\boldsymbol{x}}_{h_{[2]}}} &= -\frac{\partial \boldsymbol{h}_{\alpha\beta_{[1]}}}{\partial \hat{\boldsymbol{x}}_{n_{[2]}}} = \frac{-\,sin(\varphi_o)\,sin(\lambda_o)\,sin(\hat{\alpha}) - cos(\lambda_o)cos(\hat{\alpha})}{s\big[cos(\hat{\beta})\big]} \\
\frac{\partial \boldsymbol{h}_{\alpha\beta_{[1]}}}{\partial \hat{\boldsymbol{x}}_{h_{[3]}}} &= -\frac{\partial \boldsymbol{h}_{\alpha\beta_{[1]}}}{\partial \hat{\boldsymbol{x}}_{n_{[3]}}} = \frac{cos(\varphi_o)\,sin(\hat{\alpha})}{s\big[cos(\hat{\beta})\big]} \\
\frac{\partial \boldsymbol{h}_{\alpha\beta_{[2]}}}{\partial \hat{\boldsymbol{x}}_{h_{[1]}}} &= -\frac{\partial \boldsymbol{h}_{\alpha\beta_{[2]}}}{\partial \hat{\boldsymbol{x}}_{n_{[1]}}} = \frac{-s[cos(\varphi_o)\,cos(\lambda_o)] + sin(\hat{\beta})\Delta x}{s^2\big[cos(\hat{\beta})\big]} \\
\frac{\partial \boldsymbol{h}_{\alpha\beta_{[2]}}}{\partial \hat{\boldsymbol{x}}_{h_{[2]}}} &= -\frac{\partial \boldsymbol{h}_{\alpha\beta_{[2]}}}{\partial \hat{\boldsymbol{x}}_{n_{[2]}}} = \frac{-s[cos(\varphi_o)\,sin(\lambda_o)] + sin(\hat{\beta})\Delta y}{s^2\big[cos(\hat{\beta})\big]} \\
\frac{\partial \boldsymbol{h}_{\alpha\beta_{[2]}}}{\partial \hat{\boldsymbol{x}}_{h_{[3]}}} &= -\frac{\partial \boldsymbol{h}_{\alpha\beta_{[2]}}}{\partial \hat{\boldsymbol{x}}_{n_{[3]}}} = \frac{-s[sin(\varphi_o)] + sin(\hat{\beta})\Delta z}{s^2\big[cos(\hat{\beta})\big]}
\end{aligned} \tag{4-46}$$

As a function of the changing system state, the vector function ***h***, and its partial derivatives that make up the measurement matrix ***H***, must be computed at each time step.

4.2.6.3 Measurement Noise

The measurement noise matrix is a square matrix that expresses the variance of the measurements being applied to the filter. Again, the contents of the matrix vary based on the type of measurement being applied. Equation (4-47) defines the measurement noise matrix for a position measurement, (4-48) defines the matrix for a velocity measurement, and (4-49) provides the matrix for AOA/AOE measurements.

$$\boldsymbol{R}_{pos} = \widetilde{\boldsymbol{R}}_{r_n} \tag{4-47}$$

Where: $\boldsymbol{R}_{pos}$ = Measurement uncertainty for position measurements (3x3).

$\widetilde{\boldsymbol{R}}_{r_n}$ = Navigation aid ECEF position covariance matrix (3x3). Refer to (4-10).

$$\boldsymbol{R}_{vel} = \widetilde{\boldsymbol{R}}_{\dot{r}_n} \tag{4-48}$$

Where: $\boldsymbol{R}_{vel}$ = Measurement uncertainty for velocity measurements (3x3).

$\widetilde{\boldsymbol{R}}_{\dot{r}_n}$ = Navigation aid ECEF velocity covariance matrix (3x3). Refer to (4-13).

$$\boldsymbol{R}_{\alpha\beta} = \begin{bmatrix} (\hat{\sigma}_{\alpha}^{N68})^2 & 0 \\ 0 & \left(\hat{\sigma}_{\beta}^{N68}\right)^2 \end{bmatrix} \tag{4-49}$$

Where: $\boldsymbol{R}_{\alpha\beta}$ = Measurement uncertainty for AOA and AOE measurements (2x2).

$(\hat{\sigma}_{\alpha}^{N68})^2$ = Variance of navigation aid AOA measurement. Refer to (4-50).

$\left(\hat{\sigma}_{\beta}^{N68}\right)^2$ = Variance of navigation aid AOE measurement. Refer to (4-50).

4.2.6.3.1 Navigation Aid AOA/AOE Measurement Uncertainty

The theoretical and empirical accuracy of AOA measurements from ADS-B data was discussed in Section 3.1.1. As described, Reck et al. determined that an AOA RMSE of 0.66° was achievable when the AOA was calculation was made from ADS-B capable aircraft operating at a variety of ranges [7]. Based on this empirical result, 0.7° has been selected as the AOA and AOE measurement uncertainty. The resulting AOA and AOE measurement variance is presented in (4-50).

$$(\hat{\sigma}_{\alpha}^{N68})^2 = \left(\hat{\sigma}_{\beta}^{N68}\right)^2 = \left(0.7\frac{\pi}{180}\right)^2 \quad \textbf{(4-50)}$$

Where: $(\hat{\sigma}_{\alpha}^{N68})^2$ = Estimated variance of navigation aid AOA measurement in radians.

$\left(\hat{\sigma}_{\beta}^{N68}\right)^2$ = Estimated variance of navigation aid AOE measurement in radians.

4.2.6.4 Kalman Gain

In keeping with the Schmidt-Kalman philosophy of compensating for the unobservable navigation aid bias states, the optimal EKF Kalman gain for use in the covariance update equations is computed per (2-25). The symbolic results of this expression are denoted in (4-51). The sub-optimal Kalman gain, described in Section 2.3.2, is then established for use in the state update equations as shown in (4-52).

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_h \\ \boldsymbol{K}_{n_1} \\ \vdots \\ \boldsymbol{K}_{n_m} \end{bmatrix} \quad \textbf{(4-51)}$$

Where: $\boldsymbol{K}$ = Optimal EKF Kalman gain, computed using (2-25).

$\boldsymbol{K}_h$ = Kalman gain for the host vehicle.

$\boldsymbol{K}_{n_m}$ = Kalman gain for the m^{th} navigation aid.

$= \begin{bmatrix} \boldsymbol{K}_{n_e} \\ \boldsymbol{K}_{n_n} \end{bmatrix}$

$\boldsymbol{K}_{n_e}$ = Kalman gain for the essential navigation aid states.

$\boldsymbol{K}_{n_n}$ = Kalman gain for the nuisance navigation aid states.

$$\boldsymbol{K}_s = \boldsymbol{K} \mid \boldsymbol{K}_{n_n} = 0 \quad \textbf{(4-52)}$$

4.2.7 Filter Initialization

The navigation algorithm is dependent upon the ability to receive ADS-B data and its associated AOA/AOE information from multiple airborne navigation aids. Conceptually, the system does not require *a-priori* knowledge of the host vehicle position because suitable information is readily available to determine an initial estimate. Accordingly, the filter is initialized following the calculation of the initial host vehicle position estimate. Subsequently, the filtering algorithm is used to simultaneously track the host vehicle position and the airborne navigation aids. The present section provides an overview of the process that is used to initialize the filter's host vehicle state and the process that is used to initialize the system state for airborne navigation aids being tracked.

4.2.7.1 Host Vehicle Initialization

The host vehicle state is initialized when a complete data set has been received from two or more airborne navigation aids within the previous two seconds, and the angle created by the intersection of the measured lines of bearing from any pair of navigation aids to the host vehicle is in the range [20°, 160°]. This allowable AOI range is somewhat arbitrary and exists to ensure that a unique intersection can be computed from the lines of bearing. Recall from the spherical intersection description in Section 2.6.1.2.5 that an infinite number of intersections may exist when the point of intersection lies on the line between the two reference positions. When one also considers the uncertainty present in the AOA measurements, the uncertainty in the intersection problem becomes excessive as the point of intersection simply nears the line between the reference positions. This phenomenon is illustrated in Figure 4-1.

The shaded region of Figure 4-1A shows that the region of uncertainty for a 90° AOI is a bounded trapezoidal area. In this figure, points P and Q represent the ADS-B transmitting aircraft, while point R represents the intersection derived from the AOA measurements. Figure 4-1B illustrates that for a 25° AOI, the region of uncertainty takes on a bounded diamond shape with an extended tail. This region of uncertainty is much larger than the region of uncertainty for a 90° AOI. As one might expect, the region of uncertainty becomes unbounded as the AOI approaches 0°. This can be expressed mathematically as $\lim_{\gamma \to 0} \sigma_{r_l}(\gamma) = \infty$.

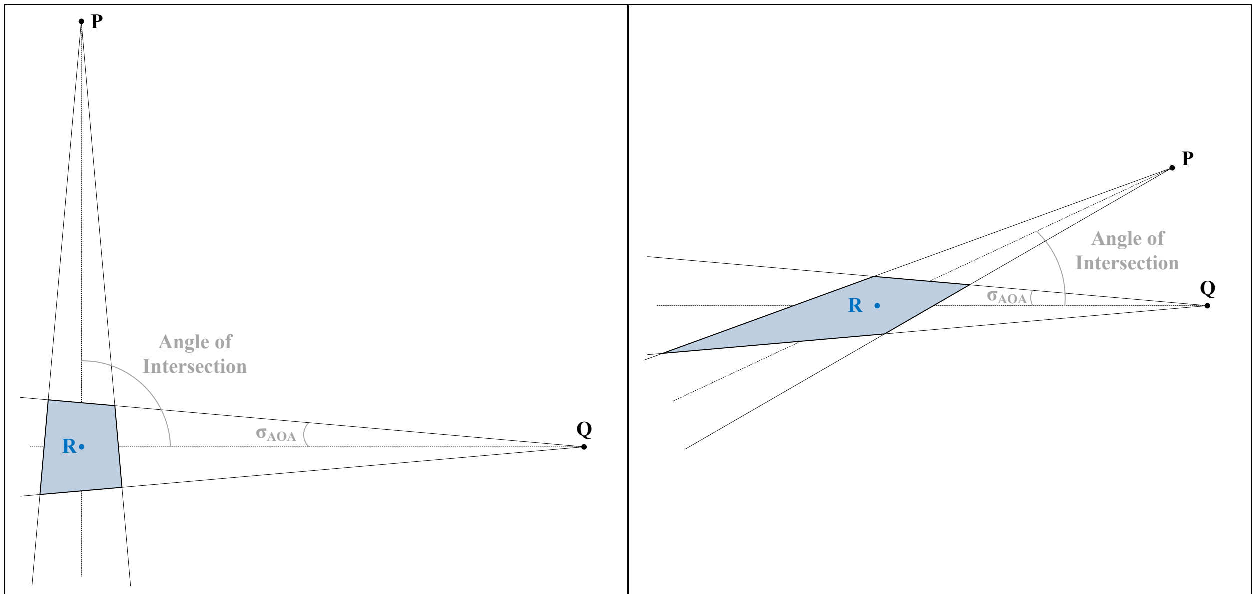

Figure 4-1A: Region of Uncertainty for a 90° Angle of Intersection.

Figure 4-1B: Region of Uncertainty for a 25° Angle of Intersection.

Figure 4-1: Region of Position Uncertainty as a Function of Angle of Intersection.

Figure 4-2 quantifies the empirical qualitative results presented in Figure 4-1 using the Ancker method that was described in Section 3.2.1. That is, Figure 4-2 presents the actual lateral position uncertainty as a function of the AOI; where both airborne navigation aids are assumed to be at the maximum 190 NM range (Section 2.1.6) and subject to an AOA uncertainty of 0.7° (Section 4.2.6.3.1). The results indicate that the lateral position uncertainty becomes asymptotic when the AOI drops below 10°. To avoid singularities and issues with sensitivity, the AOI limit of 20° has been selected.

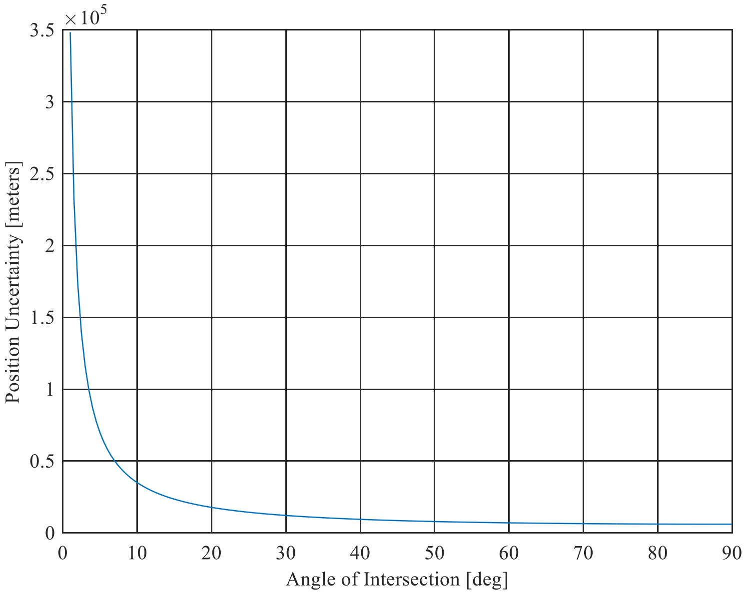

Figure 4-2: Lateral Position Uncertainty as a Function of Angle of Intersection.

4.2.7.1.1 *Host Vehicle State Vector Initialization*

The initial position of the host vehicle is computed using the Spherical Great Circle Intersection equations described in Section 2.6.1.2.5 and the Spherical Altitude equations presented in Section 2.6.1.2.6. The inputs to these equations will be the two airborne navigation aid positions that satisfy the AOI constraint and whose AOI is nearest to 90°, the reciprocal of the corresponding AOA measurements, and the AOE to each navigation aid. This results in the 3-dimensional host vehicle position expressed in geographic latitude/longitude coordinates. These results are then rotated into the ECEF frame using the methods presented in Section 2.6.1.1.4. Once expressed in the ECEF frame, the values are directly assigned to the host vehicle state vector, with the velocity and acceleration states initialized to zero as shown in (4-53).

$$\boldsymbol{x}_{h_0} = \begin{bmatrix} \hat{r}_{h_x} & \hat{r}_{h_y} & \hat{r}_{h_z} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{4-53}$$

Where: $\boldsymbol{x}_{h_0}$ = Initial host vehicle aid state vector.

$\hat{r}_{h_x}$ = Estimated host vehicle ECEF position x coordinate.

$\hat{r}_{h_y}$ = Estimated host vehicle ECEF position y coordinate.

$\hat{r}_{h_z}$ = Estimated host vehicle ECEF position z coordinate.

4.2.7.1.2 Host Vehicle State Covariance Initialization

The initial lateral variance of the host vehicle position is computed using the Ancker method detailed in Section 3.2.1. The inputs to this procedure are: the estimated host vehicle position, the two airborne navigation aid positions used to determine the host vehicle's initial position, the airborne navigation aid lateral variance (augmented to account for latency), and the variance of the AOA measurements.

As described in Section 4.2.7.1, the host vehicle initialization process requires that the positions used during initialization have been received in the previous two seconds. In addition to this two second latency, the ADS-B data itself is subject to a maximum of 0.6 seconds of latency (refer to Section 2.1.1.2.2). Finally, the navigation algorithm processing rate of 100 milliseconds introduces an additional data latency. To account for the fact that the airborne navigation aid positions used in this calculation were not instantaneously received prior to the calculation, the navigation aid lateral position variance is augmented to account for the possible position latency described above. The position latency variances are based on the aircraft reported velocity and the median latency time. This process is expressed in (4-54), noting that the resulting uncertainty is expressed as a lateral uncertainty in the NED frame, not in the ECEF frame.

$$\left(\hat{\sigma}_{r_{n_l}}^{N68}\right)^2 = \left(\tilde{\sigma}_{r_{n_l}}^{N68}\right)^2 + \left(\hat{\sigma}_{\Delta t_{r_{NE_0}}}^{N68}\right)^2 \tag{4-54}$$

Where: $\left(\hat{\sigma}_{r_{n_l}}^{N68}\right)^2$ = Latency compensated, navigation aid initial lateral position variance.

$\left(\tilde{\sigma}_{r_{n_l}}^{N68}\right)^2$ = Measured navigation aid lateral position variance. Refer to (4-8).

$\left(\hat{\sigma}_{\Delta t_{r_{NE_0}}}^{N68}\right)^2$ = Navigation aid lateral position latency compensation.

$= (\Delta t_0)^2 \left[\left(\tilde{\dot{r}}_{n_N}\right)^2 + \left(\tilde{\dot{r}}_{n_E}\right)^2\right]$

Δt_0 = 50th percentile position initialization measurement latency:

50th percentile initialization delay	1.00 seconds
50th percentile ADS-B transmit latency	0.20 seconds
50th percentile navigation processing latency	0.05 seconds
Total	1.25 seconds

$\tilde{\dot{r}}_{n_N}$ = ADS-B reported navigation aid velocity north.

$\tilde{\dot{r}}_{n_E}$ = ADS-B reported navigation aid velocity east.

The Anker calculations are then used determine the host vehicle lateral position variance in an arbitrary *X, Y* frame. This process is expressed as a general function in (4-55).

$$\left(\boldsymbol{\hat{\sigma}}_{r_{h_{x,y}}}^{N68}\right)^2 = f\left[\hat{r}_h, \hat{r}_{n_1}, \hat{r}_{n_2}, \left(\hat{\sigma}_{r_{n_{l_1}}}^{N68}\right)^2, \left(\hat{\sigma}_{r_{n_{l_2}}}^{N68}\right)^2, (\sigma_{\alpha}^{N68})^2\right] \tag{4-55}$$

Where: $\left(\boldsymbol{\hat{\sigma}}_{r_{h_{x,y}}}^{N68}\right)^2$ = Initial host vehicle lateral position variance in an arbitrary X,Y frame.

$\hat{r}_h$ = Estimated host vehicle position in geographic coordinates.

$\hat{r}_{n_{1,2}}$ = Estimated navigation aid *m* position in geographic coordinates.

$\left(\hat{\sigma}_{r_{n_{l_{1,2}}}}^{N68}\right)^2$ = Estimated navigation aid *m* lateral position variance (4-54).

$(\sigma_{\alpha}^{N68})^2$ = Angle of arrival measurement variance (4-50).

The initial host vehicle lateral position uncertainty is then taken to be the maximum of the X, Y uncertainty values as shown in (4-56). This establishes a conservative, normally distributed, estimate of the host vehicle lateral position uncertainty.

$$\left(\hat{\sigma}_{r_{h_l}}^{N68}\right)^2 = max\left[\left(\hat{\sigma}_{r_{h_x}}^{N68}\right)^2 \quad \left(\hat{\sigma}_{r_{h_y}}^{N68}\right)^2\right] \tag{4-56}$$

Where: $\left(\hat{\sigma}_{r_{h_l}}^{N68}\right)^2$ = Estimated variance of initial host vehicle lateral position.

The initial vertical uncertainty for the host vehicle considers the vertical error incited by the AOE measurement, the navigation aid's vertical position uncertainty, and the vertical error induced through horizontal coupling. This horizontal coupling error is an expression of the vertical error that is a direct result of the uncertainty in the host vehicle and navigation aid lateral position estimates. An expression for the initial host vehicle vertical uncertainty is given in (4-57), with the individual terms detailed in (4-58) through (4-60).

$$\left(\hat{\sigma}_{r_{h_v}}^{N68}\right)^2 = \left(\hat{\sigma}_{r_{h_{v_\beta}}}^{N68}\right)^2 + \left(\hat{\sigma}_{r_{n_v}}^{N68}\right)^2 + \left(\sigma_{r_{h_{v_{hc}}}}^{N68}\right)^2 \tag{4-57}$$

Where: $\left(\hat{\sigma}_{r_{h_v}}^{N68}\right)^2$ = Estimated variance of initial host vehicle vertical position.

$\left(\hat{\sigma}_{r_{h_{v_\beta}}}^{N68}\right)^2$ = Host vehicle vertical position variance due to AOE measurement error (4-58).

$\left(\hat{\sigma}_{r_{n_v}}^{N68}\right)^2$ = Latency compensated, navigation aid vertical position variance (4-59).

$\left(\sigma_{r_{h_{v_{hc}}}}^{N68}\right)^2$ = Host vehicle vertical position variance due to horizontal coupling (4-60).

$$\left(\hat{\sigma}_{r_{h_{v_\beta}}}^{N68}\right)^2 = s_{hn}^2\left(\hat{\sigma}_\beta^{N68}\right)^2 \tag{4-58}$$

Where: $\left(\hat{\sigma}_{r_{h_{v_\beta}}}^{N68}\right)^2$ = Host vehicle vertical position variance due to AOE measurement error.

s_{hn}^2 = Squared spherical distance from the host vehicle to the navigation aid.

$\left(\hat{\sigma}_\beta^{N68}\right)^2$ = Angle of elevation measurement variance (4-50).

$$\left(\hat{\sigma}_{r_{n_v}}^{N68}\right)^2 = \left(\tilde{\sigma}_{r_{n_v}}^{N68}\right)^2 + \left(\hat{\sigma}_{\Delta t_{r_{D_0}}}^{N68}\right)^2 \tag{4-59}$$

Where: $\left(\hat{\sigma}_{r_{n_v}}^{N68}\right)^2$ = Estimated, latency compensated, navigation aid vertical position variance.

$\left(\tilde{\sigma}_{r_{n_v}}^{N68}\right)^2$ = Measured navigation aid vertical position variance (4-9).

$\left(\hat{\sigma}_{\Delta t_{r_{D_0}}}^{N68}\right)^2$ = Navigation aid vertical position latency compensation.

= $(\Delta t_0)^2\left(-\tilde{\dot{r}}_{n_U}\right)^2$

Δt_0 = 50th percentile position initialization measurement latency. Refer to (4-54).

$\tilde{\dot{r}}_{n_U}$ = ADS-B reported navigation aid velocity up.

$$\left(\sigma_{r_{h_{v_{hc}}}}^{N68}\right)^2 = tan^2(\tilde{\beta})\left[\left(\hat{\sigma}_{r_{h_l}}^{N68}\right)^2 + \left(\hat{\sigma}_{r_{n_l}}^{N68}\right)^2\right] \tag{4-60}$$

Where: $\left(\sigma_{r_{h_{v_{hc}}}}^{N68}\right)^2$ = Host vehicle vertical position variance due to horizontal coupling error.

$\tilde{\beta}$ = Measured angle of elevation from the host vehicle to the navigation aid.

$\left(\hat{\sigma}_{r_{h_l}}^{N68}\right)^2$ = Estimated variance of initial host vehicle lateral position. Refer to (4-56).

$\left(\hat{\sigma}_{r_{n_l}}^{N68}\right)^2$ = Estimated variance of navigation aid lateral position. Refer to (4-54).

The host vehicle lateral and vertical uncertainty values are then combined into an NED vector and rotated in to the ECEF frame using the familiar process shown in (4-61).

$$\boldsymbol{P}_{r_{h_0}} = \boldsymbol{C}_{NED}^{ECEF}\left(diag\begin{bmatrix} \left(\hat{\sigma}_{r_{h_l}}^{N68}\right)^2 \\ \left(\hat{\sigma}_{r_{h_l}}^{N68}\right)^2 \\ \left(\hat{\sigma}_{r_{h_v}}^{N68}\right)^2 \end{bmatrix}\right) \boldsymbol{C}_{NED}^{ECEF^T} \tag{4-61}$$

Where: $\boldsymbol{P}_{r_{h_0}}$ = Estimated covariance of initial host vehicle position in ECEF (3x3).

$\boldsymbol{C}_{NED}^{ECEF}$ = Coordinate conversion matrix from NED to ECEF = $\boldsymbol{C}_{ECEF}^{NED^T}$

$diag$ = Operator to express column vector as diagonal matrix.

$\left(\hat{\sigma}_{r_{h_l}}^{N68}\right)^2$ =Estimated variance of initial host vehicle lateral position. Refer to (4-56)

$\left(\hat{\sigma}_{r_{h_v}}^{N68}\right)^2$ =Estimated variance of initial host vehicle vertical position. Refer to (4-57).

Finally, the covariance values for the host vehicle states are established as indicated in (4-62).

$$\boldsymbol{P}_{h_0} = \begin{bmatrix} \boldsymbol{P}_{r_{h_0}} & 0 & 0 \\ 0 & \boldsymbol{P}_{\dot{r}_{h_0}} & 0 \\ 0 & 0 & \boldsymbol{P}_{\ddot{r}_{h_0}} \end{bmatrix} \tag{4-62}$$

Where: $\boldsymbol{P}_{h_0}$ = Initial host vehicle state covariance (9x9).

$\boldsymbol{P}_{r_{h_0}}$ =Estimated covariance of initial host vehicle position in ECEF (3x3). (4-61)

$\boldsymbol{P}_{\dot{r}_{h_0}}$ =Estimated covariance of initial host vehicle velocity in ECEF (3x3).

$$= diag \begin{bmatrix} (5.1444\ m/s)^2 \\ (5.1444\ m/s)^2 \\ (5.1444\ m/s)^2 \end{bmatrix}$$

$\boldsymbol{P}_{\ddot{r}_{h_0}}$ = Estimated covariance of initial host vehicle acceleration in ECEF (3x3).

$$= 0.1\boldsymbol{P}_{\dot{r}_{h_0}}$$

4.2.7.2 Airborne Navigation Aid Initialization

Prior to utilizing a newly detected airborne navigation aid in the filter, a complete set of ADS-B data must be received. This requires that at least one each of an airborne position message (Section 2.1.1.2.2), airborne velocity message (Section 2.1.1.2.3), and airborne operational status message (Section 2.1.1.2.4) be received from the ADS-B transmitting aircraft. Ideally, based on the message transmission rates detailed in Section 2.1.3, a complete ADS-B data set could be received within a 0.5-second window. However, due to the possibility of ADS-B transmission collisions, a 2-second window is implemented to ensure that the ADS-B message set is not only complete, but also relatively current.

Following host vehicle initialization and upon receipt of a complete and current ADS-B data set, the system state vector (Section 4.2.1) and state covariance matrix (Section 4.2.3) are expanded by 12 elements to accommodate the newly acquired navigation aid.

4.2.7.2.1 Airborne Navigation Aid State Vector Initialization

The initial position and velocity elements of the new state vector elements are initialized to the position and velocity data received in the ADS-B message set. Recall that ADS-B position and velocity values are not presented in the ECEF frame and must be rotated into the ECEF frame using the techniques presented in Sections 4.2.4.1 and 4.2.4.2. In keeping with the Schmidt-Kalman filtering techniques, the state vector values for the position and velocity measurement bias states are initialized to zero. The ensuing initial state vector for an arbitrary navigation aid is given in (4-63).

$$\boldsymbol{x}_{n_0} = \begin{bmatrix} \tilde{r}_{n_x} & \tilde{r}_{n_y} & \tilde{r}_{n_z} & \tilde{\dot{r}}_{n_x} & \tilde{\dot{r}}_{n_y} & \tilde{\dot{r}}_{n_z} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \qquad \textbf{(4-63)}$$

Where: $\boldsymbol{x}_{n_0}$ = Initial navigation aid state vector.

$\tilde{r}_{n_x}$ = Measured navigation aid ECEF x coordinate. Refer to (4-4).

$\tilde{r}_{n_y}$ = Measured navigation aid ECEF y coordinate. Refer to (4-5).

$\tilde{r}_{n_z}$ = Measured navigation aid ECEF z coordinate. Refer to (4-6).

$\tilde{\dot{r}}_{n_x}$ = Measured navigation aid ECEF velocity x. Refer to (4-7).

$\tilde{\dot{r}}_{n_y}$ = Measured navigation aid ECEF velocity y. Refer to (4-7).

$\tilde{\dot{r}}_{n_z}$ = Measured navigation aid ECEF velocity z. Refer to (4-7).

4.2.7.2.2 Airborne Navigation Aid State Covariance Initialization

The initial airborne navigation aid state covariance matrix is set to the position and velocity uncertainty values received in the ADS-B message. This process is necessary to convert the ADS-B uncertainty values into the ECEF frame, to express the uncertainty values as 1-sigma, normally distributed variables, and to compensate for the initialization latency. The initial airborne navigation aid state covariance matrix is presented as (4-64).

$$\boldsymbol{P}_{n_0} = \begin{bmatrix} \widehat{\boldsymbol{P}}_{r_{n_0}} & 0 & 0 & 0 \\ 0 & \widetilde{\boldsymbol{R}}_{\dot{r}_n} & 0 & 0 \\ 0 & 0 & 4\widehat{\boldsymbol{P}}_{r_{n_0}} & 0 \\ 0 & 0 & 0 & 4\widetilde{\boldsymbol{R}}_{\dot{r}_n} \end{bmatrix} \quad \textbf{(4-64)}$$

Where: $\boldsymbol{P}_{n_0}$ = Initial navigation aid state covariance.

$\widehat{\boldsymbol{P}}_{r_{n_0}}$ = Estimated covariance of initial navigation aid position in ECEF (3x3).

$= \widetilde{\boldsymbol{R}}_{r_n} + \widehat{\boldsymbol{P}}_{\Delta t_0}$

$\widetilde{\boldsymbol{R}}_{r_n}$ = Navigation aid ECEF position measurement covariance (3x3). Refer to (4-10).

$\widetilde{\boldsymbol{R}}_{\dot{r}_n}$ = Navigation aid ECEF velocity measurement covariance (3x3). Refer to (4-13).

$\widehat{\boldsymbol{P}}_{\Delta t_0}$ = Navigation aid initial position latency compensation matrix (3x3).

$$= \left(\boldsymbol{C}_{NED}^{ECEF}\right) \left(diag \begin{bmatrix} (\Delta t_0)^2 \left[\left(\tilde{\dot{r}}_{n_N}\right)^2 + \left(\tilde{\dot{r}}_{n_E}\right)^2 \right] \\ (\Delta t_0)^2 \left[\left(\tilde{\dot{r}}_{n_N}\right)^2 + \left(\tilde{\dot{r}}_{n_E}\right)^2 \right] \\ (\Delta t_0)^2 \left(-\tilde{\dot{r}}_{n_U}\right)^2 \end{bmatrix} \right) \left(\boldsymbol{C}_{NED}^{ECEF}\right)^T$$

$\left(\boldsymbol{C}_{NED}^{ECEF}\right)$ = Coordinate conversion matrix from NED to ECEF.

Δt_0 = Navigation aid latency for the initialization process. Refer to (4-54).

$\tilde{\dot{r}}_{n_N}$ = ADS-B reported navigation aid velocity north.

$\tilde{\dot{r}}_{n_E}$ = ADS-B reported navigation aid velocity east.

$\tilde{\dot{r}}_{n_U}$ = ADS-B reported navigation aid velocity up.

4.2.8 Reset Navigation Aid Bias State Covariance on Accuracy Change

Recall that the bias state covariance values are estimated as Gauss-Markov random stochastic processes using the Schmidt-Kalman philosophy. In this configuration, the bias states themselves are not estimated, but the effects of the bias uncertainty are accounted for in the covariance matrix through cross-correlation. The white-noise correlation time constants associated with the Gauss-Markov bias state estimates result in a slow response to changes in navigation aid reported accuracy. Therefore, the bias state covariance is reinitialized whenever the reported accuracy associated with a navigation aid changes. That is, when the NACp or GVA reported by a navigation aid changes, the navigation aid position bias covariance values are reinitialized. Likewise when the NACv changes, the navigation aid velocity bias covariance values are reinitialized. The remainder of this section details these reinitializations.

The 12 x 12 covariance matrix associated with a particular navigation aid can be segmented as shown in (4-65). The 6 x 6 submatrix $\boldsymbol{P}_{n_\delta}$ can be further sub-divided into the 3 x 3 matrices as shown in (4-66).

$$\boldsymbol{P}_n = \begin{bmatrix} \boldsymbol{P}_{n_e} & \boldsymbol{P}_{n_{e\delta}} \\ \boldsymbol{P}_{n_{e\delta}} & \boldsymbol{P}_{n_\delta} \end{bmatrix} \tag{4-65}$$

Where: $\boldsymbol{P}_{n_e}$ = Navigation aid essential state covariance matrix (6x6).
$\boldsymbol{P}_{n_\delta}$ = Navigation aid bias state covariance matrix (6x6).
$\boldsymbol{P}_{n_{e\delta}}$ = Navigation aid covariance between the essential states and bias states (6x6).

$$\boldsymbol{P}_{n_\delta} = \begin{bmatrix} \boldsymbol{P}_{n_{\delta_r}} & \boldsymbol{P}_{n_{\delta_{r\dot{r}}}} \\ \boldsymbol{P}_{n_{\delta_{r\dot{r}}}} & \boldsymbol{P}_{n_{\delta_{\dot{r}}}} \end{bmatrix} \tag{4-66}$$

Where: $\boldsymbol{P}_{n_{\delta_r}}$ = Navigation aid position bias state covariance matrix (3x3).
$\boldsymbol{P}_{n_{\delta_{\dot{r}}}}$ = Navigation aid velocity bias state covariance matrix (3x3).
$\boldsymbol{P}_{n_{\delta_{r\dot{r}}}}$ = Navigation aid covariance between the position and velocity bias states (3x3).

The following reinitializations occur when either the navigation aid reported NACp or GVA changes from its previous value:

1. All off diagonal elements associated with the rows and columns of $\boldsymbol{P}_{n_{\delta_r}}$ throughout the composed $\boldsymbol{P}$ matrix are set to zero.
2. $\boldsymbol{P}_{n_{\delta_r}}$ is set equal to $4 * \boldsymbol{R}_{pos}$ where $\boldsymbol{R}_{pos}$ is defined in (4-47).

Similarly, the following reinitializations occur when the navigation aid reported NACv changes from its previous value:

1. All off diagonal elements associated with the rows and columns of $\boldsymbol{P}_{n_{\delta_{\dot{r}}}}$ throughout the composed $\boldsymbol{P}$ matrix are set to zero.
2. $\boldsymbol{P}_{n_{\delta_{\dot{r}}}}$ is set equal to $4 * \boldsymbol{R}_{vel}$ where $\boldsymbol{R}_{vel}$ is defined in (4-48).

4.2.9 Removing Airborne Navigation Aids from the Filter

As an airborne navigation aid moves beyond the radio horizon of the host vehicle, or otherwise stops producing ADS-B transmission, the reception of ADS-B messages from that navigation aid will cease. When an ADS-B message has not been received from a navigation aid in the previous 20 seconds, the navigation aid is removed from the filter as discussed in Section 4.1. Removal of a navigation aid from the filter requires that all corresponding state vector elements be removed and the state vector collapsed. Likewise, all corresponding rows and columns of the covariance matrix are removed and the matrix collapsed.

4.2.10 Estimated Position Uncertainty

The lateral and vertical EPU values for the host vehicle and for each tracked airborne navigation aid are established from the filter's state covariance matrix. The lateral EPU is expressed as a Rayleigh random variable at the 95% containment level as shown in (4-67); while (4-68) provides the vertical EPU (VEPU) expressed as a Gaussian random variable, also at the 95% containment level. Equation (4-69) provides the mechanization used to express the host vehicle's ECEF state covariance values in the NED frame as required to carry out the calculations in (4-67) and (4-68). The final host vehicle lateral EPU is then smoothed using a first order low pass filter with a 60 second time constant. This filtering action applies only to decreasing EPU values, increases in EPU are applied instantaneously. This mechanization ensures that the final EPU is a conservative estimate of the position accuracy.

A similar operation is performed for the navigation aid EPU calculations by making the appropriate substitution to utilize covariance matrix values associated with the particular navigation aid's position estimate rather than the covariance values associated with the host vehicle's position estimate. The final navigation aid EPU output is not filtered simply because this is not considered an essential output of the filter.

$$\hat{\sigma}_{h_{r_l}}^{R95} = 2.4477\sqrt{max\left[\left(\hat{\sigma}_{h_{r_N}}^{N68}\right)^2 \quad \left(\hat{\sigma}_{h_{r_E}}^{N68}\right)^2\right]} \tag{4-67}$$

Where: $\hat{\sigma}_{h_{r_l}}^{R95}$ = Lateral estimated position uncertainty for the host vehicle (Rayleigh 95%).

$\left(\hat{\boldsymbol{\sigma}}_{h_{r_{NED}}}^{N68}\right)^2$ = Host vehicle position variance vector in NED, computed in (4-69).

2.4477 = Statistical conversion from a Gaussian 68% to a Rayleigh 95% distribution.

$$\hat{\sigma}_{h_{r_v}}^{N95} = 1.96\sqrt{\left(\hat{\sigma}_{h_{r_D}}^{N68}\right)^2} \tag{4-68}$$

Where: $\hat{\sigma}_{h_{r_v}}^{R95}$ = Vertical estimated position uncertainty for the host vehicle (Gaussian 95%).

$\left(\hat{\boldsymbol{\sigma}}_{h_{r_{NED}}}^{N68}\right)^2$ = Host vehicle position variance vector in NED, computed in (4-69).

1.96 = Statistical conversion from a Gaussian 68% to a Gaussian 95% distribution.

$$\boldsymbol{P}_{r_{h_{NED}}} = (\boldsymbol{C}_{ECEF}^{NED})(\boldsymbol{P}_{r_h})(\boldsymbol{C}_{ECEF}^{NED})^T \tag{4-69}$$

Where: $\boldsymbol{P}_{r_{h_{NED}}}$ = Host vehicle position covariance matrix in NED (3x3).

$\boldsymbol{C}_{ECEF}^{NED}$ = Coordinate conversion matrix from ECEF to NED.

$\boldsymbol{P}_{r_h}$ = Host vehicle position covariance in ECEF (3x3).

$$= \begin{bmatrix} \boldsymbol{P}_{[1,1]} & \boldsymbol{P}_{[1,2]} & \boldsymbol{P}_{[1,3]} \\ \boldsymbol{P}_{[2,1]} & \boldsymbol{P}_{[2,2]} & \boldsymbol{P}_{[2,3]} \\ \boldsymbol{P}_{[3,1]} & \boldsymbol{P}_{[3,2]} & \boldsymbol{P}_{[3,3]} \end{bmatrix} \text{ where } \boldsymbol{P} \text{ is the state covariance matrix.}$$

4.3 Algorithmic Process

This section describes the algorithmic steps that have been implemented to compute the host vehicle position from received ADS-B data using the method described in Section 4.2. An overview of the process is provided in Figure 4-3. With respect to this figure, process flow is denoted by solid connecting arrows, while data flow is denoted with dashed connecting arrows. It is clear from Figure 4-3 that the algorithm is broken into three major processing steps that follow the classic paradigm of: Input Data, Process Data, and then Output Results. The main processing loop is periodic and configured to execute at a 10 Hz rate.

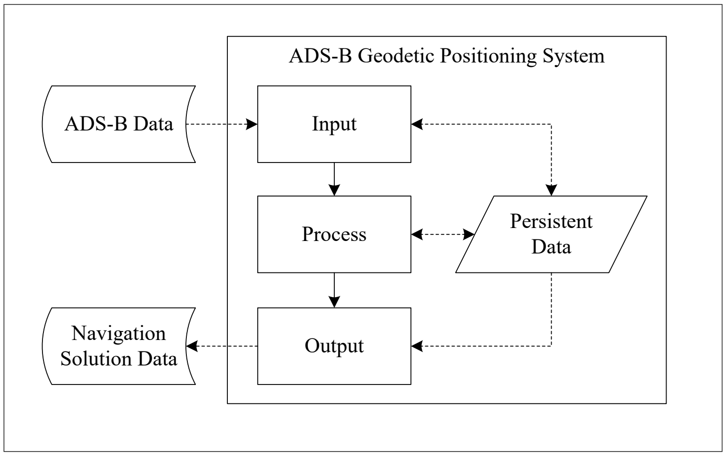

Figure 4-3: High Level Algorithmic Process.

4.3.1 ***Input Function***

The Input function collects the received ADS-B data from shared memory and saves the information into a persistent one-dimensional array for use by the Process and Output functions. Each element of the array contains a record of data applicable to a particular navigation aid; Table 4-1 through Table 4-5 detail the elements of this record structure. It should be noted that in addition to providing storage for the received ADS-B data, this record also provides persistent storage for the navigation algorithm. As such, the input function does not populate every element in this data store.

Due to the fact that the array is persistent, it must be searched each time new ADS-B data becomes available to determine if the navigation aid is already represented in the array. If the navigation aid already exists in the array, then the persistent data is overwritten by the newly received ADS-B data. If the navigation aid does not exist in the array, then the array is expanded to accommodate the newly acquired navigation aid.

The Input function then assigns a validity indication to the navigation aid. A navigation aid is considered valid for use by the algorithm when at least one of each type of ADS-B message has been received, and the NACp, NACv, and GVA values are all non-zero.

During initial development and evaluation of the algorithm it was discovered that configuring the filter to utilize NACv values higher than one (velocity uncertainty less than 3 meters/second) would occasionally cause the filter's estimate of the navigation aid positions to diverge to the point of filter instability. To prevent this divergence, the algorithm has been configured to assume a NACv value of 1 for all navigation aids, regardless of the received NACv value. It is considered future work to diagnose and correct this issue.

Table 4-1: Airborne Navigation Aid ADS-B Data Record.

Variable	Data Type	Description
adsbPos	ADSBPositionMsgType	Received ADS-B position data Table 4-2.
adsbVel	ADSBVelocityMsgType	Received ADS-B velocity data Table 4-3.
adsbStatus	ADSBStatusMsgType	Received ADS-B operational status data Table 4-4.
filterIndex	Integer	Index to this navigation aid's first state vector element.
icao	Integer	This navigation aid's ICAO address.
lastRxTime	Double	The last time a message was received for this navigation aid.
resetPosBias	Boolean	Indicates that the position bias states for this navigation aid need to be reset.
resetVelBias	Boolean	Indicates that the velocity bias states for this navigation aid need to be reset.
uncertainty	EstimatedUncertaintyType	Computed uncertainty information for this navigation aid Table 4-5.
valid	Boolean	Indicates that the navigation aid is valid for use.

Table 4-2: Airborne Navigation Aid ADS-B Position Message Data Record.

Variable	Data Type	Description
aoa	Double	Angle of arrival for this message.
aoe	Double	Angle of elevation for this message.
rxTime	Double	The time that this message was received.
lat	Double	The received ADS-B position latitude.
lon	Double	The received ADS-B position longitude.
alt	Double	The received ADS-B position altitude.

Table 4-3: Airborne Navigation Aid ADS-B Velocity Message Data Record.

Variable	Data Type	Description
aoa	Double	Angle of arrival for this message.
aoe	Double	Angle of elevation for this message.
rxTime	Double	The time that this message was received.
velE	Double	The received ADS-B east velocity.
velN	Double	The received ADS-B north velocity.
velU	Double	The received ADS-B vertical velocity.
nacV	Integer	The received ADS-B NACv value.
sigmaVel	Double	The 1σ normally distributed velocity uncertainty determined from the NACv value.

Table 4-4: Airborne Navigation Aid ADS-B Operational Status Message Data Record.

Variable	Data Type	Description
aoa	Double	Angle of arrival for this message.
aoe	Double	Angle of elevation for this message.
rxTime	Double	The time that this message was received.
nacP	Integer	The received ADS-B NACp value.
gva	Integer	The received ADS-B GVA value.
sigmaLateral	Double	The 1σ normally distributed lateral position uncertainty determined from the NACp value.
sigmaVertical	Double	The 1σ normally distributed vertical position uncertainty determined from the GVA value.

Table 4-5: Airborne Navigation Aid Estimated Uncertainty Data Record.

Variable	Data Type	Description
EPU	Double	The estimated lateral position uncertainty expressed as a 95% containment Rayleigh random variable.
EVU	Double	The estimated lateral velocity uncertainty expressed as a 95% containment Rayleigh random variable.
VEPU	Double	The estimated vertical position uncertainty expressed as a 95% containment Gaussian random variable.
VEPU	Double	The estimated vertical velocity uncertainty expressed as a 95% containment Gaussian random variable.

4.3.2 Process Function

The Process function implements the EKF and associated operations detailed in Section 4.2 and its subsections. A block diagram showing the process flow between the major subfunctions is given as Figure 4-4. Because the major subfunctions in Figure 4-4 closely implement the logic described in Section 4.2, with the exception of the Apply Measurements block, little additional detail is provided here.

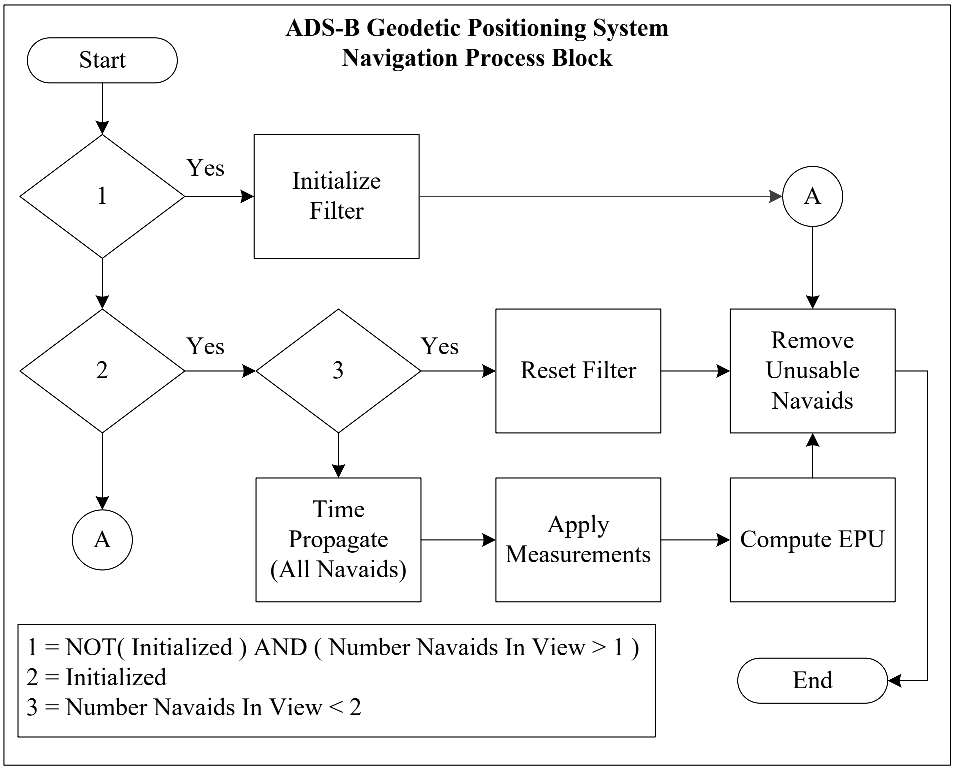

Figure 4-4: Navigation Process High Level Overview.

The Apply Measurements block implements the observation update portion of the EKF described in Section 4.2.6. As a germane portion of the filter, a block diagram of the process flow is provided in Figure 4-5. As shown in this figure, the Apply Measurements block loops over each valid navigation aid contained in the navigation aid array populated by the Input function (refer to Section 4.3.1) to determine if fresh ADS-B data has been received from the navigation aid. If fresh data is available, then the filter is examined to determine if the particular navigation aid is already a member of the filter. If it is not, then as described in Section 4.2.7.2, the filter is expanded to accommodate the new navigation aid. If the navigation aid is already included in the filter, then the fresh data is applied to filter using the hierarchy and techniques described in Section 4.2.6

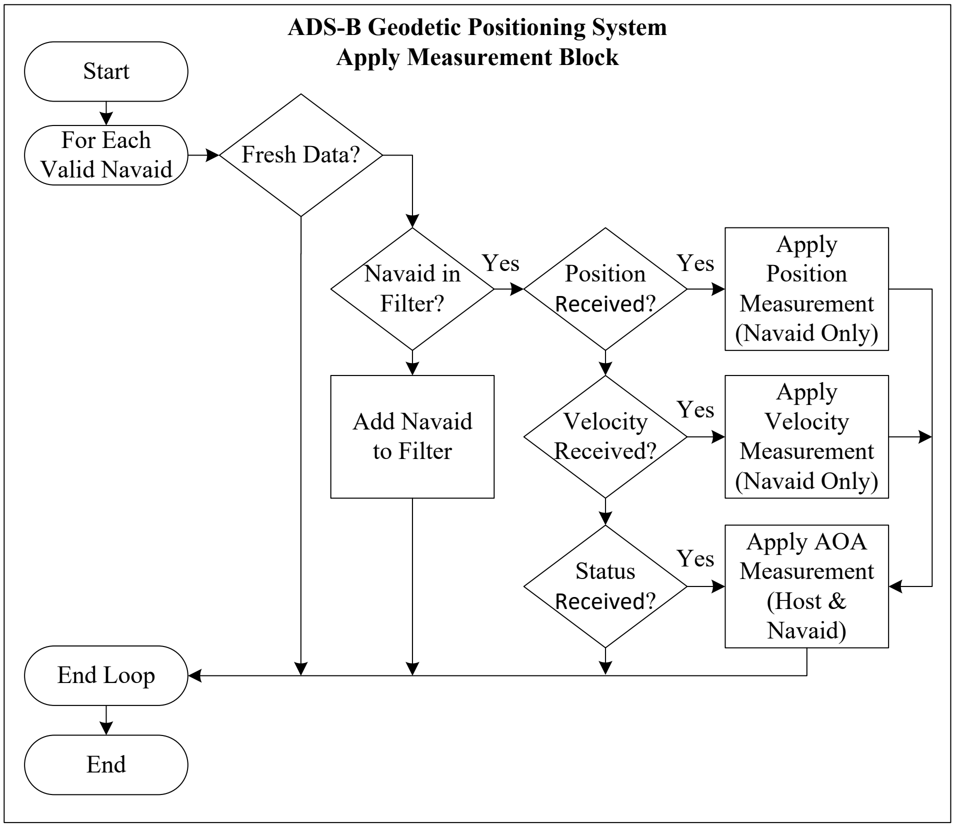

Figure 4-5: Apply Measurement High Level Overview.

4.3.3 Output Function

The Output function assembles the computational results from the Process function, formats the data for publication, and saves the information to shared memory for use by external applications. The host vehicle data is published as a record containing the information given in Table 4-6. In addition to this data, a count of the number of navigation aids in use is also published. The navigation aid data is published as an array of records, where each array element contains the data record given in Table 4-6, populated with information specific to a particular navigation aid.

Table 4-6: Navigation Solution Output Data Record.

Variable	Data Type	Description
icao	Integer	The vehicle's ICAO address.
lat	Double	The latitude of the position estimate.
lon	Double	The longitude of the position estimate.
alt	Double	The altitude of the position estimate.
velN	Double	The vehicle's north velocity.
velE	Double	The vehicle's east velocity.
velD	Double	The vehicle's vertical velocity.
EPU	Double	The estimated lateral position uncertainty expressed as a 95% containment Rayleigh random variable.
EVU	Double	The estimated lateral velocity uncertainty expressed as a 95% containment Rayleigh random variable.
VEPU	Double	The estimated vertical position uncertainty expressed as a 95% containment Gaussian random variable.
VEPU	Double	The estimated vertical velocity uncertainty expressed as a 95% containment Gaussian random variable.
valid	Boolean	Indicates if the vehicle's navigation data is valid.

5.0 SIMULATION

The method detailed in this paper represents a theoretical algorithmic approach to determine a host vehicle's geodetic position based on AOA and AOE measurements of ADS-B data. Although both Reck's and Faragher's research have demonstrated the ability to determine AOA from ADS-B data, all analysis performed to validate this method has been conducted using simulations rather than real world data. To that end, a simulation was developed that emulates ADS-B capable aircraft to provide pseudo ADS-B data to the navigation algorithm. The term pseudo ADS-B data is used because the data content available from the simulation does not strictly follow the bit-level ADS-B protocol; however, the data is consistent in content, precision, and transmission rate. This method has been selected to emulate the data that would likely be provided by an ADS-B AOA/AOE receiver should one be constructed.

The simulation consists of two separate software applications: an ADS-B simulation library and a test driver. These mutually dependent applications are the subject of the following two subsections. A demonstration of the simulation is provided in the final subsection of this topic.

5.1 ADS-B Simulation Library

The ADS-B simulation library is a software library that was developed in C# to simulate ADS-B capable aircraft. The simulation allows for the host vehicle and multiple airborne navigation aids to be configured, each with a user defined flight profile and corresponding navigation accuracy. Control of the simulation, and configuration of each aircraft's flight profile, is accomplished through the use of an Application Program Interface (API) that can be called by the test harness. The exposed API commands are described in APPENDIX I. Once configured, the simulation interfaces provide access to the true position and velocity information for all defined aircraft, as well as the AOA, AOE, and pseudo ADS-B data for those navigation aids that are considered to be in range of the host vehicle.

When an aircraft is added to the simulation, a random nominal performance model is established for the aircraft. This nominal performance model uniquely describes the turn rate, rate of climb/descent, and acceleration/deceleration rate for the aircraft. The turn rate is drawn from a uniform distribution on the interval [135, 225] degrees per minute. Note that the range of turn rates is centered on the standard aviation turn rate of 180 degrees per minute. The nominal altitude rate is drawn from a uniform distribution on the interval [2000, 6000] feet per minute. Finally, the nominal acceleration rate is drawn from a uniform distribution on the interval [2.5, 7.5] feet per second squared. When used, these nominal rates are further perturbed by additive Gaussian noise with a standard deviation of 10% of the nominal value for the acceleration and altitude rates, and a standard deviation of 5% of the nominal value for the turn rate.

The ADS-B data provided by the simulation is also subject to various perturbations including: latency, bias, and Gaussian noise. Upon initialization of each simulated aircraft, a random latency value is selected from a uniform distribution over the range [-0.6, 0.2] seconds. This latency value emulates the ADS-B position latency described in Section 2.1.1.2.2. Accordingly, it remains fixed for the duration of the simulation run and is applied to all ADS-B position reports from that particular aircraft.

An initial aircraft position and velocity bias is also drawn for each aircraft during initialization. The bias is drawn from a Rayleigh distribution with scale factor (mode) equal to the 1-sigma equivalent of the simulated aircraft's defined NACp uncertainty (NACv for velocity bias). The direction of the bias is computed as the arc tangent as of the normally distributed constituent components used to generate the Rayleigh random variable. Refer to Section 2.5.2 for an overview of the process by which a Rayleigh random variable is formed. To emulate an inflight change in GNSS accuracy due to a constellation change or other phenomena, a new bias is selected for each aircraft at a 900 second interval. In order to stagger the bias reselection between aircraft, the first bias reselection time is randomly drawn for each aircraft from a uniform distribution on the range [0, 900] seconds.

In addition to bias, the reported position and velocity values are subject to additive, zero mean, uncorrelated, disturbances generated from the same distributions as the bias values. However, the standard deviation of the noise is limited to 10% of the defined 1-sigma equivalent of the NACp or NACv value.

The AOA value reported by the simulation defines the true north referenced azimuth from the host vehicle to the navigation aid from which ADS-B data is being simulated. The AOA and AOE values are also disturbed by zero mean, uncorrelated, Gaussian noise with a standard deviations defined during simulation instantiation. Nominally, based on Reck's empirical results described in the Section 3.1.1, a standard deviation of 0.7° is selected.

Finally, the AOE provided by the simulation defines a locally level referenced elevation angle from the host vehicle to the navigation aid for which ADS-B data is being simulated. The AOE is also subject to zero mean, uncorrelated, Gaussian noise with a standard deviation defined during simulation instantiation.

5.2 ADS-B Simulation Test Harness

In an effort to enable black box testing of the ADS-B geodetic positioning algorithm, a test harness was written in MATLAB. The test harness provides simultaneous control over both the ADS-B simulation library and the navigation algorithm; it also serves as the interface between the two applications. The boundary between the simulation library and the algorithm under test ensures that the algorithm only operates on the ADS-B data received from the test harness. A block diagram of the test harness, including its interface to the simulation library and navigation algorithm, is provided in Figure 5-1. Note that the pseudo ADS-B data is the only data input to the navigation algorithm.

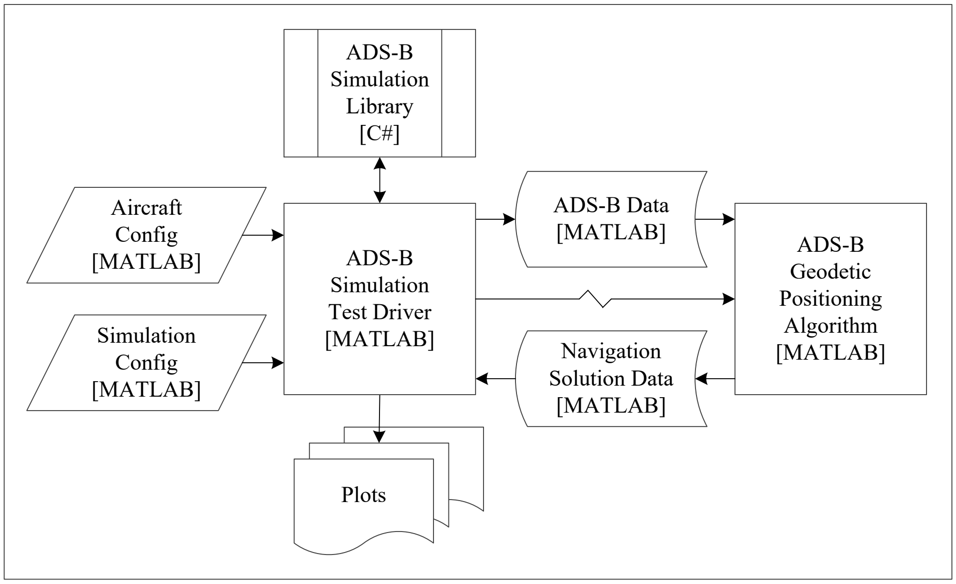

Figure 5-1: Top Level ADS-B Simulation Test Harness Block Diagram.

Control of the simulation and configuration of each aircraft's flight profile is accomplished via the use of two input configuration files, referred to as the simulation configuration file and the aircraft configuration file. The simulation configuration file is used to control the behavior of the simulation including: the length of the simulation run, the interval used to plot the results, the AOA and AOE uncertainty, and the types of real-time data plots to be generated. The aircraft configuration file is used to define the number of simulated aircraft, their respective flight plans, and their individual navigation accuracy. The aircraft configuration file is the primary customer of the simulation library's API described in APPENDIX I.

5.3 Simulation Demonstration

In order to provide an initial demonstration of the operation of the simulation, and its various perturbations, a comparison between the simulated true lateral position and the ADS-B reported lateral position for an arbitrary airborne navigation aid is presented as Figure 5-2.

Likewise, a comparison between the simulated true vertical position and the ADS-B reported vertical position for an arbitrary navigation aid is given in Figure 5-3. In order to emphasize the magnitude of the position errors, a NACp value of 2 and a GVA value of 1 were selected for this demonstration. It is clear from these plots that the aircraft is following the defined flight profile and that position noise and bias are both present. Also evident in these plots are several changes in the magnitude of the position biases. These plots provide a level of confidence that the simulation is operating as expected. Additional validation of the simulation is included in the analysis of the algorithm. This is done to confirm that the simulation provided ADS-B data that was consistent with the accuracy level configured for the particular test.

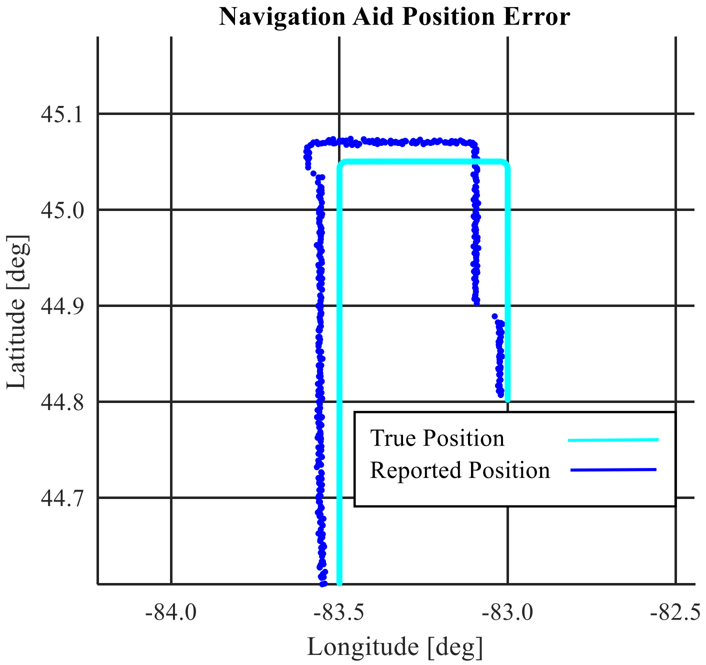

Figure 5-2: Simulated ADS-B Lateral Position Error with NACp = 2 (EPU < 4 NM) to Emphasize Position Bias Changes.

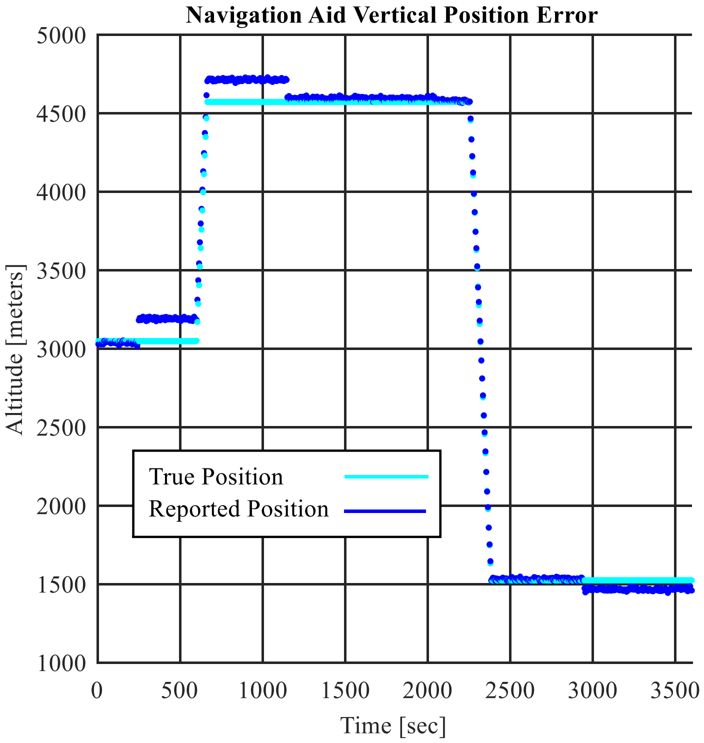

Figure 5-3: Simulated ADS-B Vertical Position Error with GVA = 1 (Altitude Error < 150 M) to Emphasize Position Bias Changes.

6.0 TEST METHOD AND EVALUATION CRITERIA

The algorithmic method described in this paper presents an unproven approach to geodetic positioning that is highly dependent on navigation aid availability, accuracy, and geometry relative to the host vehicle. Because of these limitations, there are no clear or predefined expectations for the performance or accuracy of the algorithm. Therefore, the accuracy of the geodetic positioning algorithm was evaluated by comparing the lateral EPU output by the algorithm against the measured radial position error (RPE) and the measured position uncertainty (MPU). Similar comparisons were made in the vertical dimensions between the vertical EPU (VEPU), vertical position error (VPE) and vertical MPU (VMPU).

A nearly infinite number of test scenarios could be derived to evaluate the accuracy of the navigation algorithm; however, as an initial assessment of a somewhat unique navigation solution, the test cases selected for this analysis were limited to those that represent nominal conditions. Nominal conditions are those cases where an adequate number of airborne navigation aids are available, the navigation aids are reporting positions whose errors are within the bounds of their reported uncertainty, and the navigation aids are located to provide a favorable geometry. The accuracy of the ADS-B geodetic positioning algorithm was then evaluated by performing a series of Monte-Carlo tests whose inputs spanned the range of nominal conditions.

Each of the Monte-Carlo tests was composed of a 50-run ensemble, with each run having a duration of 3600 seconds. Each of the 50-runs for a given Monte-Carlo test utilized a different seed for the random number generators utilized by the simulation. This ensured that the inputs to the algorithm under test varied between each of the runs in the ensemble. Unless otherwise specified, the lateral flight profile for the host vehicle and for all navigation aids was as shown in Figure 6-1. Similarly, the default vertical profile for the host vehicle and the airborne navigation aids is given in Figure 6-2. Every 5 seconds during each iteration of the Monte-Carlo tests, pertinent data output by the algorithm was stored to allow offline evaluation of the filter's performance.

In an effort to demonstrate the overall capability of the algorithm with a manageable number of test scenarios, three test philosophies were devised:

1. Varying number of navigation aids with predefined accuracy and flight paths.
2. Random navigation aid accuracy and flight paths.
3. Sparse parametric analysis.

The details of these test scenarios are provided in the remaining subsections of the present section. The outcome of each scenario is presented in the corresponding subsection of Section 7.0.

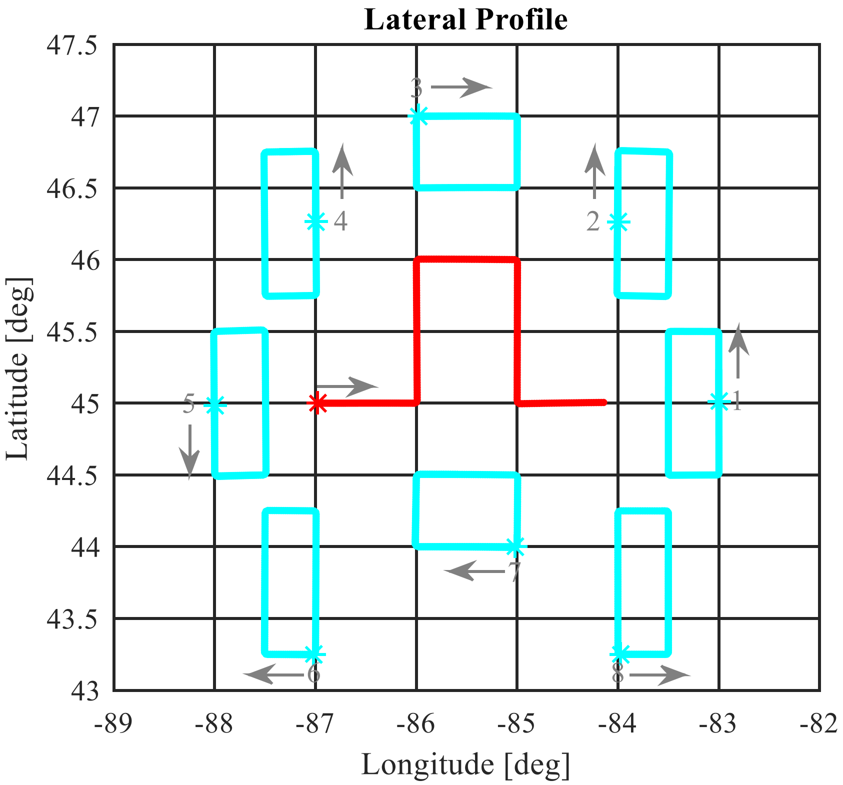

Figure 6-1: Simulated Lateral Flight Profile. Host Vehicle Path is in Red, Airborne Navigation Aid Paths are Numbered and in Cyan. Asterisks Indicate the Starting Position, with the Arrow Denoting the Initial Direction of Travel.

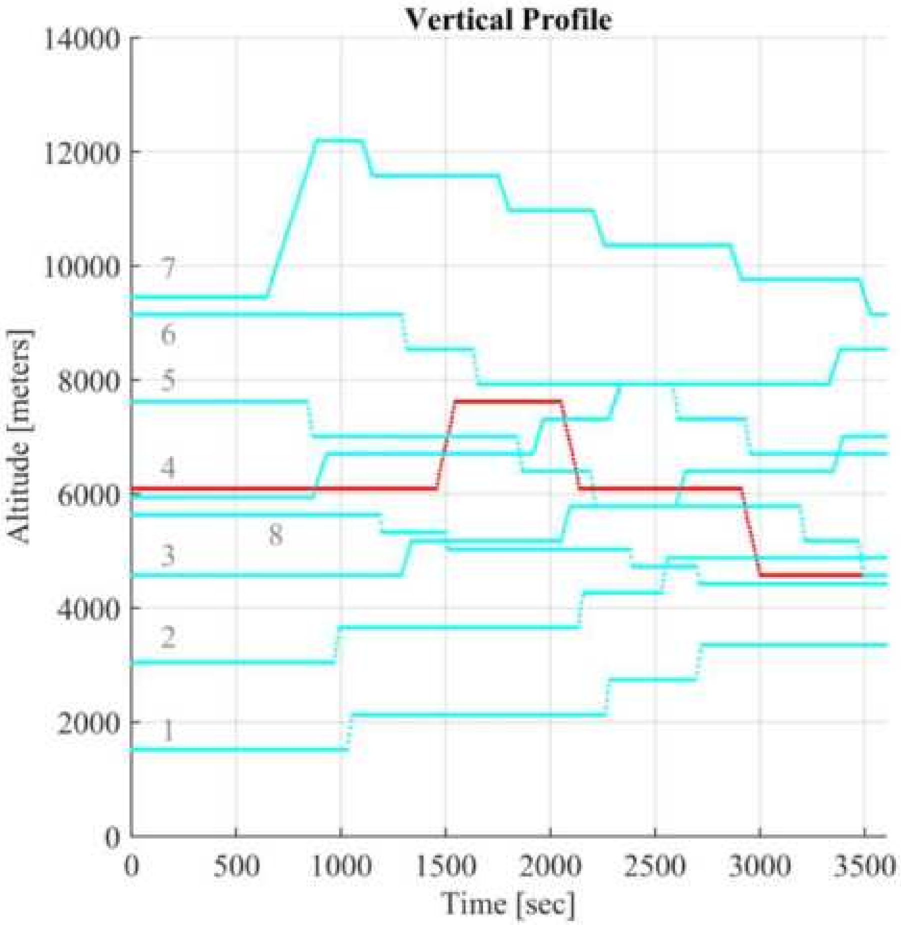

Figure 6-2: Simulated Vertical Flight Profile. Host Vehicle Path is in Red, Airborne Navigation Aid Paths are Numbered and in Cyan.

6.1 Varying Number of Airborne Navigation Aids Test Scenario

This test examined numerous capabilities of the algorithm in a single scenario. First, this scenario evaluated the algorithm's ability to self-initialize when a suitable navigation aid pair became available. Next, this scenario examined the algorithm's ability incorporate new navigation aids into the filter as they became available, and to remove navigation aids from the filter when they became invalid. Finally this scenario also allowed for evaluation of the algorithm's accuracy when exposed to varying numbers of navigation aids. This scenario was repeated three times, with different navigation aid accuracy levels used in each test. This allowed the performance of the algorithm to be examined as navigation aid accuracy diminished.

For this scenario, AOA and AOE uncertainties of 0.7° were selected; where this selection was based on Reck's work described in Section 3.1.1. Both the host vehicle and all navigation aids began in motion, following the flight profiles of Figure 6-1 and Figure 6-2. Initially, there were no navigation aids available to the algorithm, then at fixed and predetermined times, the availability of pairs of navigation aids was altered to simulate the detection of new navigation aids and/or the loss of existing navigation aids. Table 6-1 provides a listing of the available navigation aids at each inflection point during the test.

Table 6-1: Summary of Navigation Aid Availability Test Scenario 1.

Time [sec]	Navigation Aid Availability
0	None available
100	1 & 2 available
600	1, 2, 3, 4 available
1100	1, 2, 3, 4, 5, 6 available
1600	1, 2, 3, 4, 5, 6, 7, 8 available
2100	3, 4, 5, 6, 7, 8 available
2600	5, 6, 7, 8 available
3100	7 & 8 available

As stated in the introduction of the present section, this Monte-Carlo scenario was repeated three times, each with a different navigation aid accuracy as detailed in Table 6-2. The median and minimum accuracy values determined from the ADS-B accuracy survey described in Section 2.1.7 led to the selection of the accuracy values for test case one and two. Test case three was selected as a robustness scenario to evaluate the algorithm given nearly worse case inputs. The results for this series of Monte-Carlo tests are presented in Section 7.1.

Table 6-2: Navigation Aid Accuracy Settings for Monte-Carlo Testing.

Test Case	NACp	NACv	GVA
1	$10 = (\sigma_{r_l}^{R95} < 10\,m)$	$1 = (\sigma_{\dot{r}}^{R95} < 10\,m/s)$	$2 = (\sigma_{r_v}^{N95} \leq 45\,m)$
2	$6 = (\sigma_{r_l}^{R95} < 0.3\,NM)$	$1 = (\sigma_{\dot{r}}^{R95} < 10\,m/s)$	$1 = (\sigma_{r_v}^{N95} \leq 150\,m)$
3	$2 = (\sigma_{r_l}^{R95} < 4.0\,NM)$	$1 = (\sigma_{\dot{r}}^{R95} < 10\,m/s)$	$1 = (\sigma_{r_v}^{N95} \leq 150\,m)$

6.2 Random Navigation Aid Test Scenario

The intent of this test case was to evaluate the behavior of the algorithm to random inputs. In this Monte-Carlo test case, the host vehicle began in flight and traveled the host vehicle paths given in Figure 6-1 and Figure 6-2. Instead of utilizing the eight navigation aids shown in Figure 6-1 and Figure 6-2, ten navigation aids were randomly configured. Each of the ten navigation aids was initialized to a uniformly random latitude in the range [N43.0°, N47.5°], and a uniformly random longitude in the range [W82.0°, W87.0°]. The initial altitude of each navigation aid was drawn from a uniform distribution on the range [0, 9144] meters; while the initial speed was drawn from a uniform distribution over the range [100, 400] knots. Each navigation aid was programed to fly a flight path of ten randomly selected waypoints over the uniform latitude/longitude ranges given above. For every third waypoint, a new altitude and speed were selected based on the uniform altitude and speed ranges previous provided. Finally, the NACp, NACv, and GVA were all drawn from uniform integer distributions over the intervals [8, 11], [1, 4], and [1, 3] respectively. These accuracy values are based on those that could be expected to be received from actual ADS-B capable aircraft, and are justified by the ADS-B accuracy survey results presented in Section 2.1.7. Finally, 0.7° AOA and AOE uncertainty were selected for this test scenario. The results of this random test are provided in Section 7.2.

6.3 Parametric Test Scenario

The purpose of this series of tests was to systematically evaluate the performance of the filter across a wide range of possible input conditions. To accomplish this, a series of Monte-Carlo tests was performed where one parameter was altered between each test. The altered parameters included: number of navigation aids, AOA/AOE uncertainty, and navigation aid accuracy. Table 6-3 provides a listing of the input conditions used for each sub-test. The 'Navaids' column of this table indicates the navigation aid numbers from Figure 6-1 and Figure

6-2 that were used. These navigation aids are available from the beginning of the test, and remained available throughout.

Table 6-3: Input Conditions for Parametric Monte-Carlo Testing.

Test	Navaids	AOA Sigma	NACp	NACv	GVA
1	1-8	6.0°	10 $(\sigma_{r_l}^{R95} < 10\ m)$	1 $(\sigma_{\dot{r}}^{R95} < 10\ m/s)$	2 $(\sigma_{r_v}^{N95} \leq 150\ m)$
2	1-8	0.7°	10 $(\sigma_{r_l}^{R95} < 10\ m)$	1 $(\sigma_{\dot{r}}^{R95} < 10\ m/s)$	2 $(\sigma_{r_v}^{N95} \leq 150\ m)$
3	1-8	0.7°	6 $(\sigma_{r_l}^{R95} < 0.3\ NM)$	1 $(\sigma_{\dot{r}}^{R95} < 10\ m/s)$	1 $(\sigma_{r_v}^{N95} \leq 45\ m)$
4	1-8	0.7°	2 $(\sigma_{r_l}^{R95} < 4.0\ NM)$	1 $(\sigma_{\dot{r}}^{R95} < 10\ m/s)$	1 $(\sigma_{r_v}^{N95} \leq 45\ m)$
5	1-4	0.7°	10 $(\sigma_{r_l}^{R95} < 10\ m)$	1 $(\sigma_{\dot{r}}^{R95} < 10\ m/s)$	2 $(\sigma_{r_v}^{N95} \leq 150\ m)$
6	1-4	0.7°	6 $(\sigma_{r_l}^{R95} < 0.3\ NM)$	1 $(\sigma_{\dot{r}}^{R95} < 10\ m/s)$	1 $(\sigma_{r_v}^{N95} \leq 45\ m)$
7	1-4	0.7°	2 $(\sigma_{r_l}^{R95} < 4.0\ NM)$	1 $(\sigma_{\dot{r}}^{R95} < 10\ m/s)$	1 $(\sigma_{r_v}^{N95} \leq 45\ m)$
8	1-2	0.7°	10 $(\sigma_{r_l}^{R95} < 10\ m)$	1 $(\sigma_{\dot{r}}^{R95} < 10\ m/s)$	2 $(\sigma_{r_v}^{N95} \leq 150\ m)$
9	1-2	0.7°	6 $(\sigma_{r_l}^{R95} < 0.3\ NM)$	1 $(\sigma_{\dot{r}}^{R95} < 10\ m/s)$	1 $(\sigma_{r_v}^{N95} \leq 45\ m)$
10	1-2	0.7°	2 $(\sigma_{r_l}^{R95} < 4.0\ NM)$	1 $(\sigma_{\dot{r}}^{R95} < 10\ m/s)$	1 $(\sigma_{r_v}^{N95} \leq 45\ m)$

Although this systematic approach was crucial for filter tuning, it does little to demonstrate the capability of the filter beyond that which could be gleamed from the previous test scenarios. Because of this, and due to the large number of parametric test cases, a comprehensive analysis of the results was not provided. Instead, key statistics from these tests are presented in Section 7.3, with the detailed supporting data plots available in APPENDIX J.

7.0 RESULTS

In addition to presenting the results of the Monte-Carlo test scenarios described in Section 6.0, this section also provides some basic plots to demonstrate the operation of the algorithm during an arbitrary individual flight. The ensuing analysis refers to the abbreviations EPU, RPE, MPU, VEPU, VPE, and VMPU liberally; as a convenience and easy reference, they are explicitly defined here:

- Estimated Position Uncertainty (EPU) – The radius of a circle, centered on the reported lateral position, where the probability of the actual position lying inside the circle is 95% [1]. EPU is computed by the algorithm as an indication of the accuracy of the host vehicle position estimate.
- Radial Position Error (RPE) – The magnitude of the error between the reported lateral position and the true position. This is a two-dimensional measurement that is assumed to follow a Rayleigh distribution.
- Measured Position Uncertainty (MPU) – Calculated lateral position uncertainty based on the mode of the measured RPE, adjusted to the 95% containment level using the conversion constant defined in (2-65). Given an infinite number of samples, the EPU and MPU are expected to converge.
- Vertical Estimated Position Uncertainty (VEPU) – One half the distance of a line, centered on the reported vertical position, where the probability of the actual vertical position lying on the line is 95%.
- Vertical Position Error (VPE) – The difference between the reported and true vertical positions. This is a one-dimensional measurement that is assumed to follow a Gaussian distribution.

- Vertical Measured Position Uncertainty (VMPU) – Vertical position uncertainty based on the standard deviation of the measured VPE, adjusted to the 95% containment level using the conversion constant defined in (2-54). Given an infinite number of samples, the VEPU and VMPU are expected to converge.

Before presenting the aggregate results of the Monte-Carlo test scenarios that were described in Section 6.0, a couple of samples of individual (rather than Monte-Carlo) simulation runs are presented to demonstrate the basic initialization and operation of the algorithm. Although there are no quantifiable performance or accuracy expectations for this algorithm, it is generally expected that the reported position will initialize to within a reasonable proximity of the true host vehicle position, and will follow the true path of the vehicle without significant discontinuities, even when the host vehicle undergoes a change in state. To this end, Figure 7-1 provides a sample of an arbitrary initial position error, while Figure 7-2 provides a comparison between the true host vehicle position and the reported host vehicle position as the vehicle encountered a turn. These plots were configured for a one second sample rate to more faithfully capture the behavior during these interesting periods.

Figure 7-1 shows that the initial position estimate was reasonable, with an initial error on the order of 1,600 meters, followed by rapid convergence to the true position within 15 seconds. Figure 7-2 demonstrates that the reported position tracked the true position through a turn, without significant divergence from the true position of the vehicle.

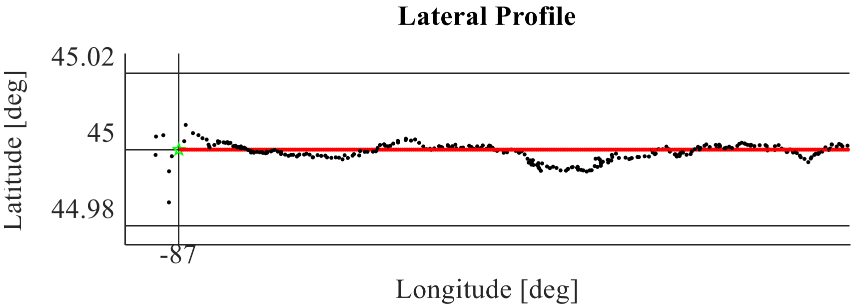

Figure 7-1: Host Vehicle Lateral Position Initialization and Lateral Path. Red Trace is the true position, Black scatter is the filter's estimate of the vehicle's position.

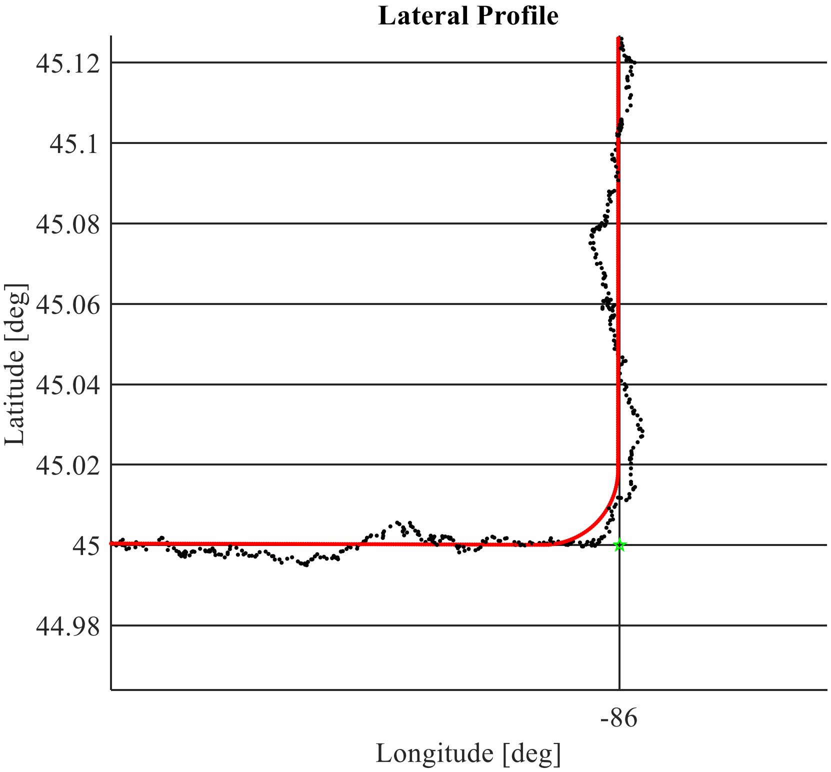

Figure 7-2: Host Vehicle Lateral Position During a Turn. Red Trace is the true position, Black scatter is the filter's estimate of the vehicle's position.

Figure 7-3 gives the estimated lateral position, true position, EPU, and RPE for the entire flight that was used to generate Figure 7-1 and Figure 7-2. Figure 7-3A provides a comparison between the estimated lateral position and the true lateral position of the host vehicle. The information in this figure suggests that the lateral position estimated by the filter tracks the true position of the host vehicle reasonably well. Figure 7-3B quantifies this assertion by providing the EPU and RPE. As expected from a position fixing algorithm that does not derive its position estimates from onboard accelerometers, the RPE exhibits significant variation from time step to time step. However, the EPU generally contains the RPE, indicating that the filter is generating a reasonable estimate of its lateral position uncertainty. For this run, the steady state EPU was on the order of 650 meters, while the RPE mode was approximately 275 meters. These values are more than sufficient for non-precision airborne navigation.

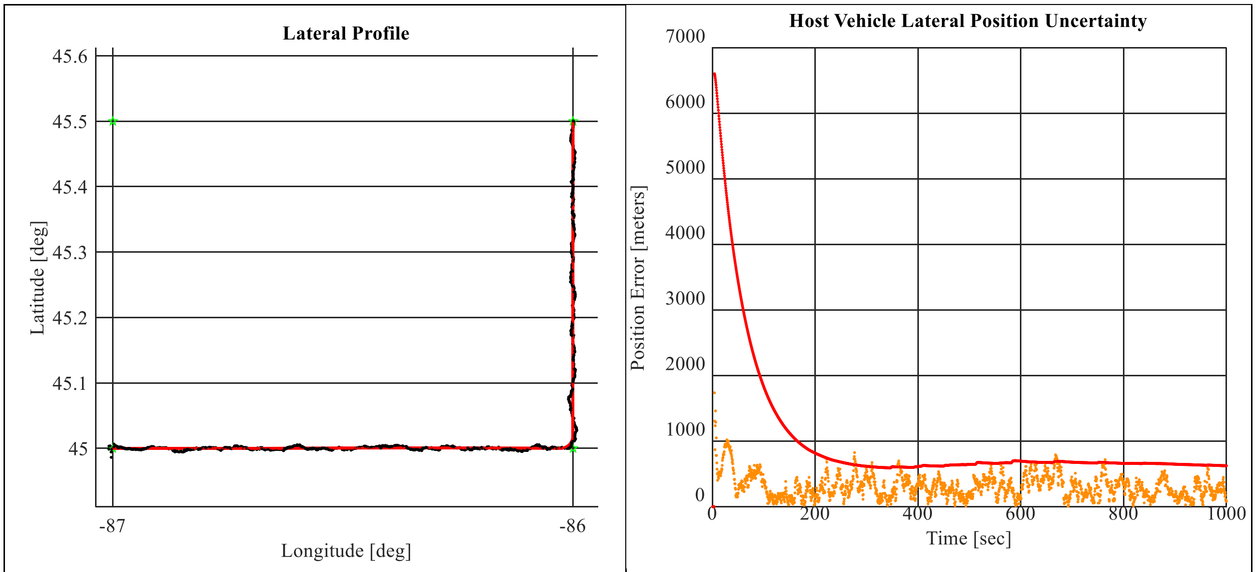

Figure 7-3A: Lateral Position Estimate (black) versus True Lateral Path (red).

Figure 7-3B: Radial Position Error (orange) and Estimated Position Uncertainty (red).

Figure 7-3: Host Vehicle Lateral Position Estimate versus True Path, Radial Position Error, and Estimated Position Uncertainty.

The estimated vertical position was compared with the true vertical position for this flight as shown in Figure 7-4A. As in the lateral case, the vertical position estimated by the filter tracks to the true vertical position with little mean divergence. That is, the mean of the error is approximately zero. Figure 7-4B validates this by providing the VEPU and VPE. It is clear from this plot that the mean error was near zero, while the mean VEPU was on the order of 350 meters. Again, these values would be adequate for non-precision navigation.

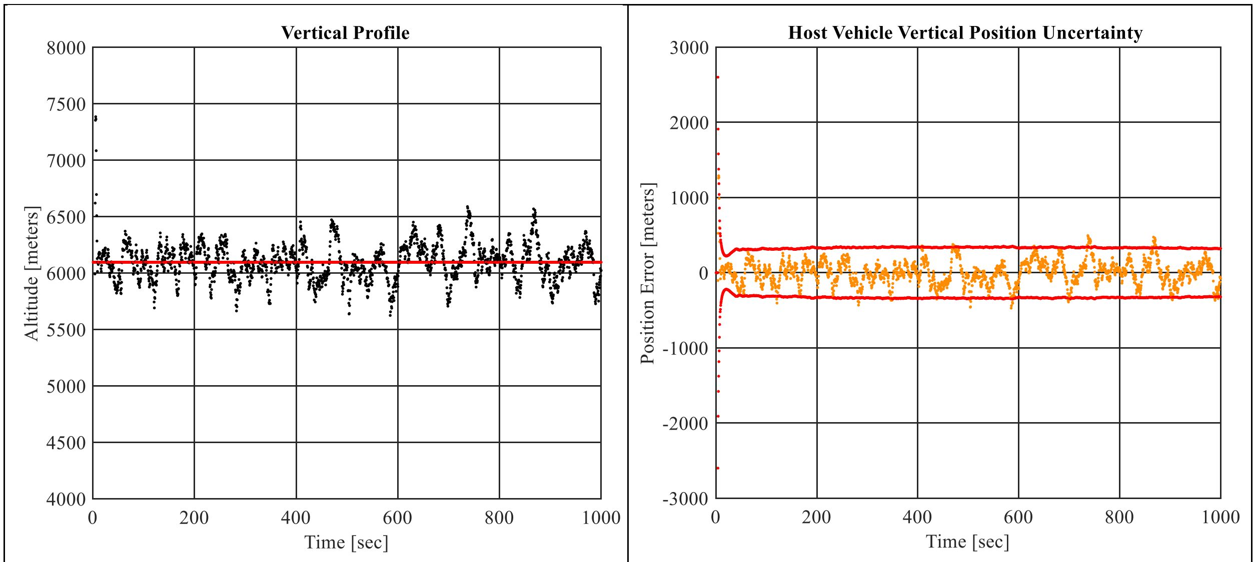

Figure 7-4A: Vertical Position Estimate (black) versus True Vertical Path (red).

Figure 7-4B: Vertical Position Error (orange) and Vertical Estimated Position Uncertainty (red).

Figure 7-4: Host Vehicle Vertical Position Estimate versus True Path, Vertical Position Error, and Vertical Estimated Position Uncertainty.

The Monte-Carlo test result analysis presented in the following subsections are numbered in such a manner as to correspond to the Section 6.0 subsection number in which the test was described. That is, the results for the test scenario described in Section 6.1 are presented in Section 7.1.

For the Variable Navigation Aid and Random Navigation Aid test scenarios, the results for each test case are presented in a series of eight data plots. These plots provide both instantaneous data and summary statistics (mean, mode, etc.). Unless otherwise specified, these statistics are computed at each time step, across all 50 runs. That is, the statistic computed at each time step has a sample size of at least 50, as illustrated in Figure 7-5. When the statistics are computed for navigation aids, the contribution from each valid navigation aid is considered in the calculation. For example, the RPE mode calculation for a test scenario with four valid navigation aids, taken at a given time step, would include 200 samples (50 Monte-Carlo runs times four navigation aids).

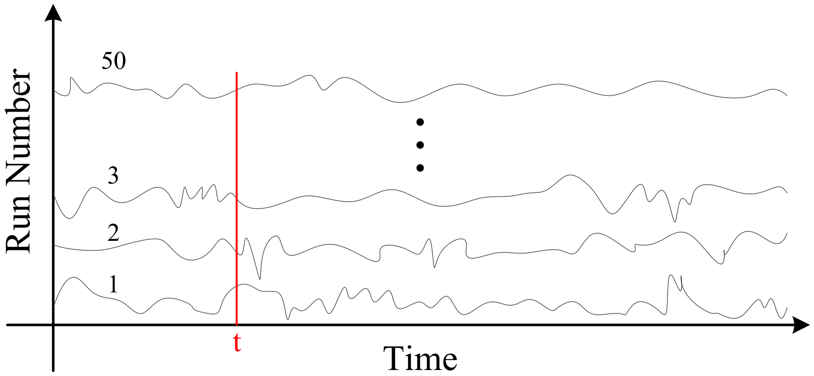

Figure 7-5: Data Statistic Calculations at Each Time Step, Across the 50 Monte-Carlo Runs.

For brevity and consistency, the purpose, content, and color encoding for each of the eight basic plots is described here:

1. **Host Vehicle EPU vs RPE** – The purpose of this plot is to evaluate the accuracy of the filter's host vehicle lateral position estimate. The blue scatter points represent the instantaneous host vehicle RPE. The red trace provides the mean of the host vehicle EPU data computed by the filter. The green trace is the Rayleigh mode computed from the RPE data. Finally, the orange trace provides the MPU computed from the RPE mode.
2. **Navigation Aid Reported EPU vs RPE** – The purpose of this plot is to validate that the ADS-B simulation is providing navigation aid lateral position data that is biased and perturbed in a manner that coincides with the selected NACp value. This ensures that the algorithm is being provided realistic lateral inputs. The red trace shows the EPU corresponding to the programmed NACp value. The blue scatter points denote the instantaneous reported navigation aid RPE for all configured navigation aids. The green trace is the Rayleigh mode computed from the instantaneous RPE data. Finally, the orange trace is the MPU computed from the RPE mode.

3. **Navigation Aid Filtered EPU vs RPE** – This plot demonstrates the algorithm's ability to filter and track the lateral ADS-B navigation aid data provided to the filter. The blue scatter points denote the instantaneous filtered navigation aid RPE for all configured navigation aids that are in range. The red trace shows the mean of all valid navigation aid EPUs calculated by the filter. The green trace is the Rayleigh mode computed from the RPE data. Finally, the orange trace is the MPU computed from the RPE mode.
4. **Navigation Aid Mean Reported RPE vs Filtered RPE** – This plot presents the mean and standard deviation for both the reported and filtered navigation aid RPE. This is provided in an effort to validate that the filter is either improving, or at least not degrading, the reported lateral position data. The blue trace provides the mean reported RPE, while the cyan trace is the standard deviation of the reported RPE. The red trace denotes the mean filtered RPE, with the magenta trace indicating the standard deviation of the filtered RPE.
5. **Host Vehicle VEPU vs VPE** – This plot is used to evaluate the accuracy of the filter's host vehicle vertical position estimate. The blue scatter points denote the instantaneous host vehicle VPE. The red trace is the mean of the host vehicle VEPU data computed by the filter. The green trace is the mean VPE, while the orange trace is the VMPU computed from the standard deviation of the VPE data.
6. **Navigation Aid Reported Vertical EPU vs VPE** – This plot is used to validate that the ADS-B simulation is providing navigation aid vertical position data that is biased and perturbed in a manner that is consistent with the selected GVA value. This ensures that the algorithm is being provided realistic vertical inputs. The red trace

shows the VEPU corresponding to the programmed GVA value. The blue scatter points denote the instantaneous reported navigation aid VPE for all configured navigation aids. The green trace is the mean VPE value computed at each time step. Finally, the orange trace is the VMPU computed from the standard deviation of the VPE data.

7. **Navigation Aid Filtered Vertical EPU vs RPE** – This plot illustrates the algorithm's ability to filter and track the vertical ADS-B navigation aid data provided to the filter. The blue scatter points represent the instantaneous filtered navigation aid VPE for all configured navigation aids that are in range. The red trace provides the mean of all valid navigation aid VEPUs calculated by the filter. The green trace is the mean VPE value computed at each time step. Finally, the orange trace is the VMPU computed from the standard deviation of the VPE data.
8. **Navigation Aid Mean Reported VPE vs Filtered VPE** – This plot presents the mean and standard deviation for both the reported and filtered navigation aid VPE. This validates that the filter is either improving, or at least not degrading, the reported vertical position data. The blue trace provides the mean reported VPE, while the cyan trace is the standard deviation of the reported VPE. The red trace denotes the mean filtered VPE, with the magenta trace showing the standard deviation of the filtered VPE.

When these plots are generated for a test case that is configured to utilize a varying number of navigation aids, vertical dashed separators are including as overlay on the plot to denote the number of navigation aids that were in use during a given time period.

As previously stated, there is no defined accuracy goal for this algorithm, but there are general expectations for the accuracy estimated provided by the filter. Because the EPU (VEPU) reflects a 95% containment level, the general expectation is that approximately 95% of all instantaneous RPE (VPE) values will fall below the EPU (VEPU) value. The exact measured containment level is provided in the title of each EPU (VEPU) plot. Because the MPU (VMPU) is a statistic computed across a finite sample size, it is not expected to identically equal the EPU (VEPU), but the MPU (VMPU) and the EPU (VEPU) statistics should be of similar magnitude. It is further expected that VPE values will exhibit a mean near zero. This expectation is not levied on RPE values because the RPE is a two-dimensional (radial) value. Finally, the expectation is that the navigation aid filtered positions will exhibit accuracy similar to that reported in the ADS-B data set.

7.1 Varying Number of Airborne Navigation Aids Test Results

This section presents the results of the Varying Number of Airborne Navigation Aids Test Scenario described in Section 6.1. As detailed in Section 6.1, the test was repeated three times with different navigation aid accuracy characteristics used for each test; accordingly, the results are presented in the three subsequent subsections.

7.1.1 Test Case 1 – NACp 10, GVA 2

The present section presents the results observed during Test Case 1 of the Varying Number of Airborne Navigation Aids test.

7.1.1.1 Filtered Lateral Performance for the Host Vehicle

The host vehicle lateral performance observed during Test Case 1 is shown in Figure 7-6. Recall that there were no navigation aids available to the algorithm for the first 100 seconds of the flight, upon which time navigation aids 1 and 2 of Figure 6-1 became available. Once this navigation aid pair was available, the filter self-initialized its position and uncertainty using the methods described in Section 4.2.7.

It is clear from Figure 7-6 that the lateral accuracy of the algorithm – reflected in both the EPU and RPE – is a function of the number of navigation aids that are available to the filter. Distinct step changes in the EPU are evident when the number of available navigation aids changes, with the minimum EPU of 511 meters occurring when eight navigation aids were in use. Likewise, the minimum observed RPE mode of 148 meters occurred when the full complement of navigation aids were available; while the maximum RPE mode of 1,902 meters occurred during initialization. The mean of the EPU was 1,355 meters, and the mean of the RPE mode was 367 meters.

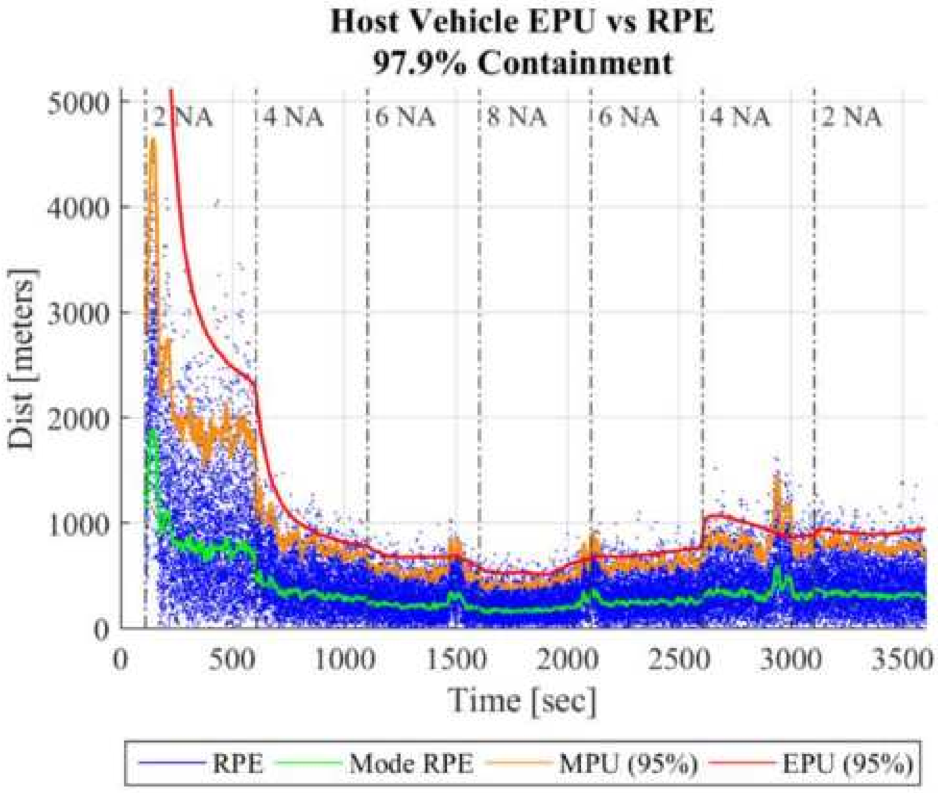

Figure 7-6: Host Vehicle Lateral Performance for the Varying Number of Navigation Aids Test with Navigation Aid Accuracies Setting of NACp 10 and GVA 2.

The more subtle effect of navigation aid geometry relative to the host vehicle is also evident in Figure 7-6. Two navigation aids are available in the intervals (100, 600] and (3100, 3600]; however, the mode of the RPE during the first interval was on the order of 900 meters, while the mode of the RPE during the second interval was on the order of 300 meters. There are two main causes for this difference in accuracy: filter convergence and navigation aid geometry relative to the host vehicle.

The filter covariance matrix is intentionally initialized to conservative values. This ensures that the filter is initially pessimistic about its own state, initially resulting in a heavy weighting of measured data. This pessimism persists for a minute or more following initialization. During this time, the filter is ill equipped to reject excessive measurement noise, resulting in a somewhat less accurate solution until the filter stabilizes.

The primary cause of the accuracy improvement in the second interval with two navigation aids was the geometry of the navigation aids relative to the host vehicle. During the first interval, the navigation aids were on the order of 270 kilometers from the host vehicle, and their AOA measurements over the interval result in an average AOI with the host vehicle of approximately 34°. This constitutes a usable, but less than ideal geometry. During the second interval, the navigation aids were roughly 130 kilometers away, and had an average AOI of 93°. Because the navigation aids were closer to the host vehicle, and the AOI was nearly at the ideal AOI of 90°, the host vehicle position uncertainty was significantly lower over the second interval, even though both intervals utilized the same number of navigation aids.

By definition, the EPU is expected to represent a 95% probability of containing the actual position. The results indicate that the mean EPU contained the 97.9% of all observed RPE. As expected, the mean EPU completely contained the mode of the RPE for this sample of 50 runs.

Also of note, the mean EPU was of similar magnitude as the MPU and contained the MPU except for a few transient cases. This is significant because the MPU is taken to be the best estimate of the true position uncertainty because it is calculated from the measured data. Ideally, in the limit as the number of runs in the Monte-Carlo test approaches infinity, the EPU and MPU would converge. These results, along with the measured 97.9% RPE containment, indicate that the filter's estimate of its lateral position uncertainty was conservative.

7.1.1.2 Characteristics of the Simulated Navigation Aid Lateral ADS-B Data

Figure 7-7 shows the instantaneous RPE, the Rayleigh mode of the RPE, the programed EPU, and the MPU of the lateral ADS-B data that produced by the simulation and provided as input to the filter. The primary takeaway from this graph is that the simulation was providing random lateral position inputs to the algorithm, as evident by the blue scatter points. Of secondary note, are that these random inputs were somewhat bounded by the NACp selected for this test case.

For this scenario, a NACp value of 10 was chosen, which according to Table 2-6, corresponds to a 95% lateral position uncertainty (EPU) of 10 meters. Accordingly, one would expect an MPU on the order of 10 meters as well, with a mean RPE mode near 4.1 meters. However, Figure 7-7 indicates that the mean MPU was on the order of 50 meters, while the mean RPE mode was 21 meters. This resulted in the EPU containing only 20.6% of the RPE samples, instead of the expected 95%.

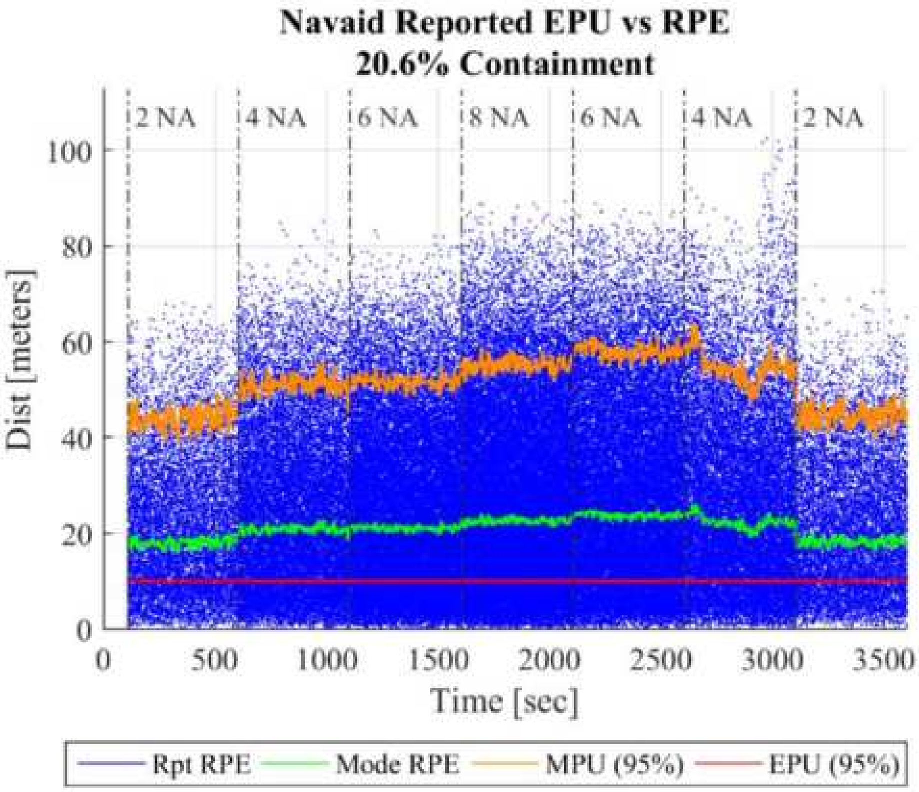

Figure 7-7: ADS-B Reported Navigation Aid Lateral Performance for the Varying Number of Navigation Aids Test with Navigation Aid Accuracy Settings of NACp 10 and GVA 2.

This seemingly excessive error is expected for navigation aids with large NACp values (low position uncertainty). This is because the navigation aid accuracy is computed by the navigation aid's onboard sensors and is reflective of the accuracy at the time of calculation. However, as described in Section 2.1.1.2.2, this navigation solution is then subject to an ADS-B transmission latency of up to 0.6 seconds. Given the speed of the airborne navigation aids, the distance traveled during the ADS-B transmission latency period can exceed the navigation aid's reported accuracy, causing the reported lateral position to appear to have more error than would otherwise be expected by the reported EPU. In other words, for highly accurate navigation aids, ADS-B latency is the dominant factor in navigation aid lateral position errors.

After considering the effects of ADS-B transmission latency, these observations suggest that the simulation was providing random lateral position reports that correspond to the programmed NACp 10 value.

7.1.1.3 Filtered Lateral Performance for the Tracked Navigation Aids

A summary graph of the filter's performance in tracking the lateral position of the airborne navigation aids is presented in Figure 7-8. The average EPU was 88.9 meters, the average RPE mode was 25.5 meters, and the average MPU was 97.8 meters. This figure shows that the filter inflated its navigation aid EPU to account for ADS-B latency, resulting in a conservative EPU that contained 97.8% of all RPE samples. Recall that the reported ADS-B EPU shown in Figure 7-7 contained only 20.6% of the reported RPE samples because it did not account for ADS-B latency. This inflation of the EPU can largely be credited to the inclusion of navigation aid bias states in the filter design.

The discontinuities in the EPU are the result of conservative covariance initialization values being used when new navigation aids are acquired. This approach ensures that measurements from a newly detected navigation aid do not immediately dominate the filter. Following initialization, the covariance values (and EPU) converge in a decaying manner in accordance with the time constant selected for the Gauss-Markov model implemented for the navigation aid bias states.

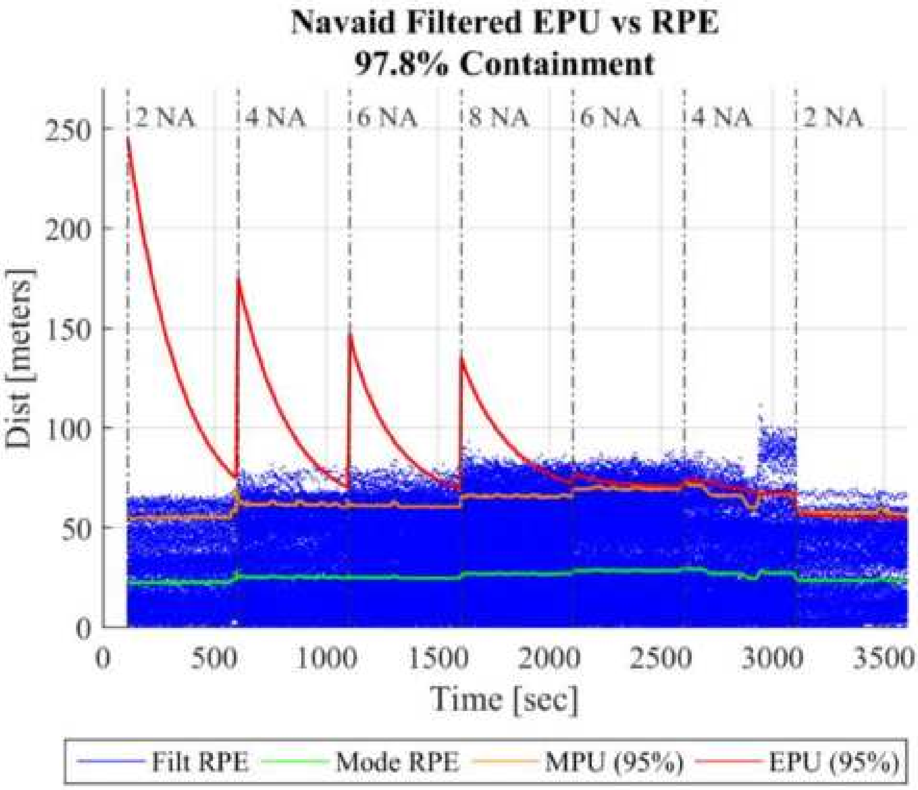

Figure 7-8: Filtered Navigation Aid Lateral Performance for the Varying Number of Navigation Aids Test with Navigation Aid Accuracy Settings of NACp 10 and GVA 2.

Figure 7-9 provides a comparison between the mean navigation aid ADS-B reported RPE and the mean navigation aid filtered RPE. The data indicates that both the mean and standard deviation of the filtered RPE values were slightly higher than the corresponding ADS-B reported RPE values. The mean reported ADS-B came in at 24.9 meters and the mean filtered RPE measured 29.8 meters. Ideally, the filtered data would exhibit a lower mean, with less standard deviation. This divergence is likely due to the assumption of linear motion that was used for time propagation of the system state (refer to Section 4.2.5) and is considered negligible for this non-precision application.

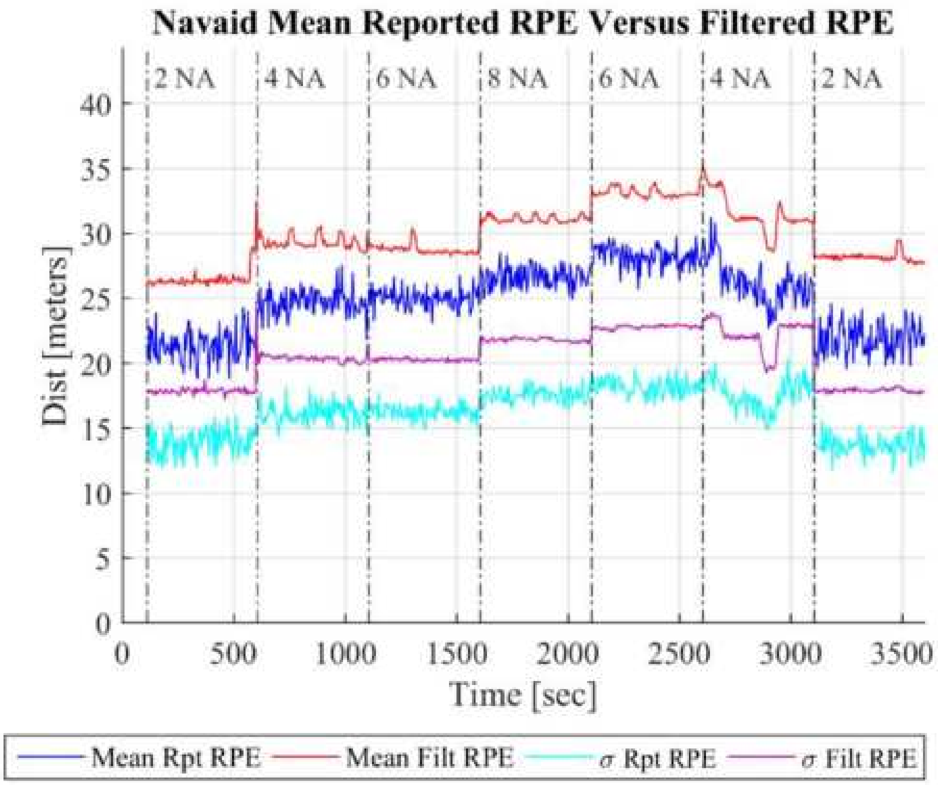

Figure 7-9: Comparison of ADS-B Reported Navigation Aid Lateral Accuracy with Filtered Navigation Aid Lateral Accuracy for the Varying Number of Navigation Aid Test with Navigation Aid Accuracy Settings of NACp 10 and GVA 2.

It is interesting to note that even though the average filtered RPE was higher than the corresponding average reported RPE, there was less variation from time step to time step in the averaged filtered data. This implies that there must also have been less variation in the instantaneous filtered position estimate. This fact is illustrated in Figure 7-10, which is a plot of the instantaneous reported RPE compared with the instantaneous filtered RPE for an arbitrary navigation aid.

Figure 7-10 clearly shows that the filtered position exhibited less variation than the associated measurement input. This is a testament to the filter's ability to reject measurement noise and produce a smoothed output. In this case, the standard deviation of the reported RPE for this navigation aid was 12.8 meters, while the corresponding standard deviation for the filtered RPE was only 4.1 meters. It should be emphasized that filtered position, in general, exhibits less variation than the reported position, but there may be cases where the level of filtering is less significant. Also note that the filtered position estimate is not always biased towards to the outer limits of the measurement data. This just happened to be the case for the navigation aid selected for this illustration.

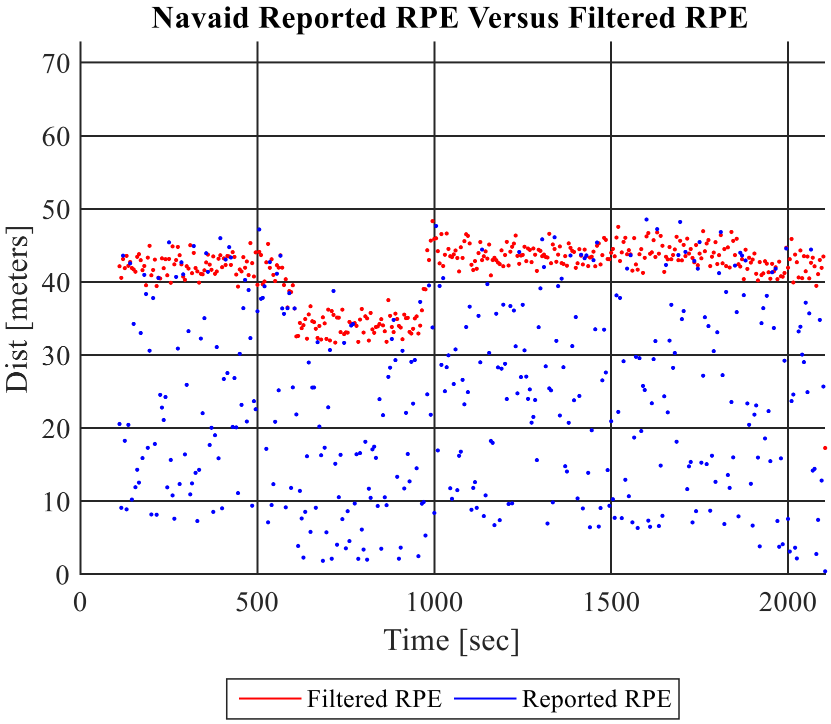

Figure 7-10: Comparison of ADS-B Reported Navigation Aid Radial Position Error with Filtered Navigation Aid Radial Position Error for the Varying Number of Navigation Aid Test with Navigation Aid Accuracy Settings of NACp 10 and GVA 2

7.1.1.4 Filtered Vertical Performance for the Host Vehicle

The vertical performance of the filter observed during Test Case 1 of the Varying Number of Airborne Navigation Aids scenario is provided in Figure 7-11. Again, there were no navigation aids available to the algorithm for the first 100 seconds of the flight, upon which time navigation aids 1 and 2 of Figure 6-2 became available. Once these navigation aids became available, the filter self-initialized its position and uncertainty using the methods described in Section 4.2.7.

Figure 7-11 reveals that the accuracy of the algorithm – reflected in both the VEPU and VPE – is also function of the number of navigation aids that are available to the filter. Distinct step changes in the VEPU are evident when the number of available navigation aids changed, with the minimum VEPU of 238 meters occurring when eight navigation aids were in use. The mean VEPU and VPE for this experiment were 466 and -4.8 meters respectively.

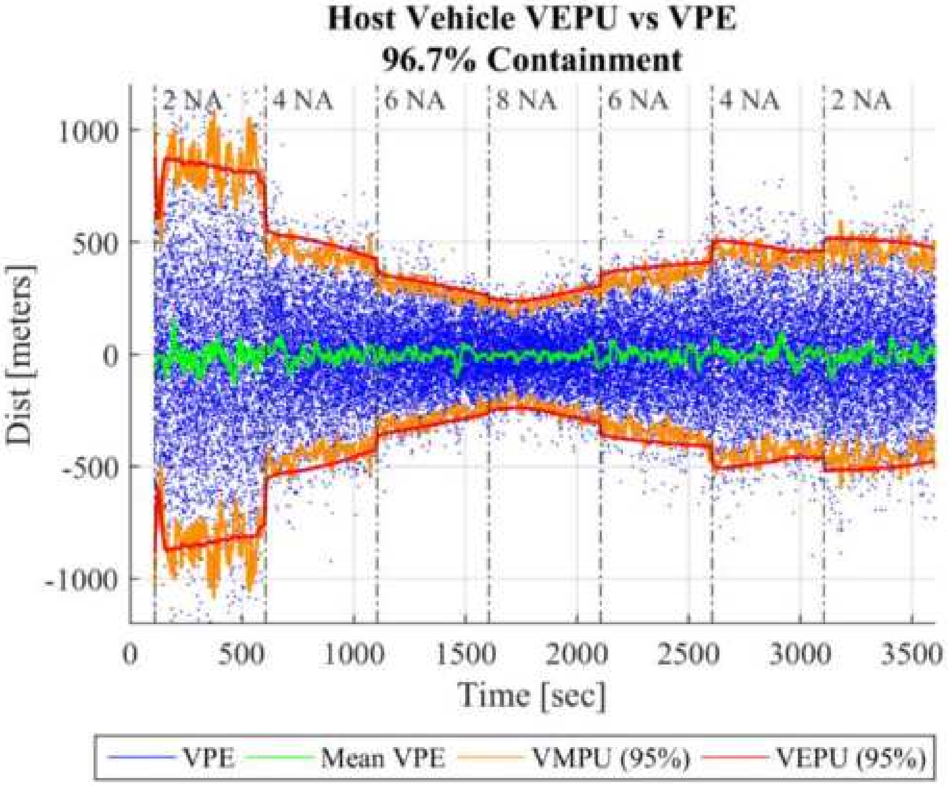

Figure 7-11: Host Vehicle Vertical Performance for the Varying Number of Navigation Aids Tests with Navigation Aid Accuracies Setting of NACp 10 and GVA 2.

Similar to the lateral performance evaluation of Section 7.1.1.1, the effects of navigation aid geometry relative to the host vehicle were again evident. During the first interval with two navigation aids available, the VEPU was on the order of 850 meters, while the VEPU during the second interval was approximately 500 meters. In this case, the primary factor contributing to the improved accuracy was proximity to the navigation aids, rather than the AOI.

Finally, Figure 7-11 indicates that the VEPU contained 96.7% of the instantaneous VPE, and was of similar magnitude to the VMPU. Given the expectation of 95% containment, the VEPU reported for this test case was also somewhat conservative.

7.1.1.5 Characteristics of the Simulated Navigation Aid Vertical ADS-B Data

Figure 7-12 provides the instantaneous VPE, the mean VPE, the programmed VEPU, and the VMPU of the vertical ADS-B data that was produced by the simulation and provided as input to the filter. This plot shows that the simulation was providing random vertical inputs to the algorithm, as denoted by the blue scatter points representing the VPE. The data also indicates that the VPE was well represented as a zero mean random variable, with a measured 95% containment level (VMPU) of 45.7 meters. This corresponds nicely with the containment level corresponding to the GVA of 2 that was selected for this test case (refer to Table 2-7). As a final check of the simulation's vertical position data, the programmed VEPU contained 94.6% of the VPE.

Dissimilar to the lateral performance analysis, the effects of ADS-B transmission latency are not evident in Figure 7-12. This is due to the fact that the navigation aids rarely undergo a change in vertical position (refer to Figure 6-2); whereas, the navigation aids are continuously experiencing a change in lateral position. Because the navigation aid vertical positions are rarely changing, the vertical position up to 0.6 seconds in the past is often equivalent to the vertical position at the time of reception. These observations suggest that the simulation is providing random vertical position reports that correspond to the programmed GVA 2 value.

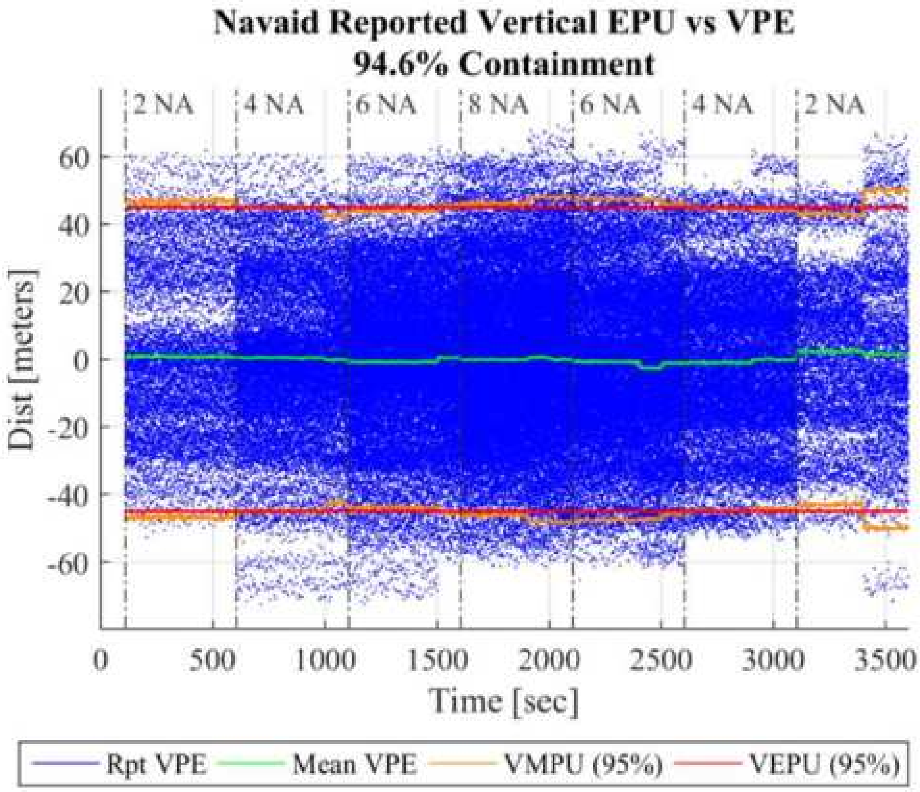

Figure 7-12: ADS-B Reported Navigation Aid Vertical Performance for the Varying Number of Navigation Aids Test with Navigation Aid Accuracy Settings of NACp 10 and GVA 2.

7.1.1.6 Filtered Vertical Performance for the Tracked Navigation Aids

A plot demonstrating the filter's performance in tracking the vertical position of the airborne navigation aids is given as Figure 7-13. The average VEPU was 53.4 meters, the average VPE was -3.7 meters, and the average VMPU was 45.7 meters. The filtered VEPU contained 98.2% of the instantaneous VPE and entirely contained the MPU. This implies that the filter again generated a conservative accuracy estimate for its tracked navigation aids.

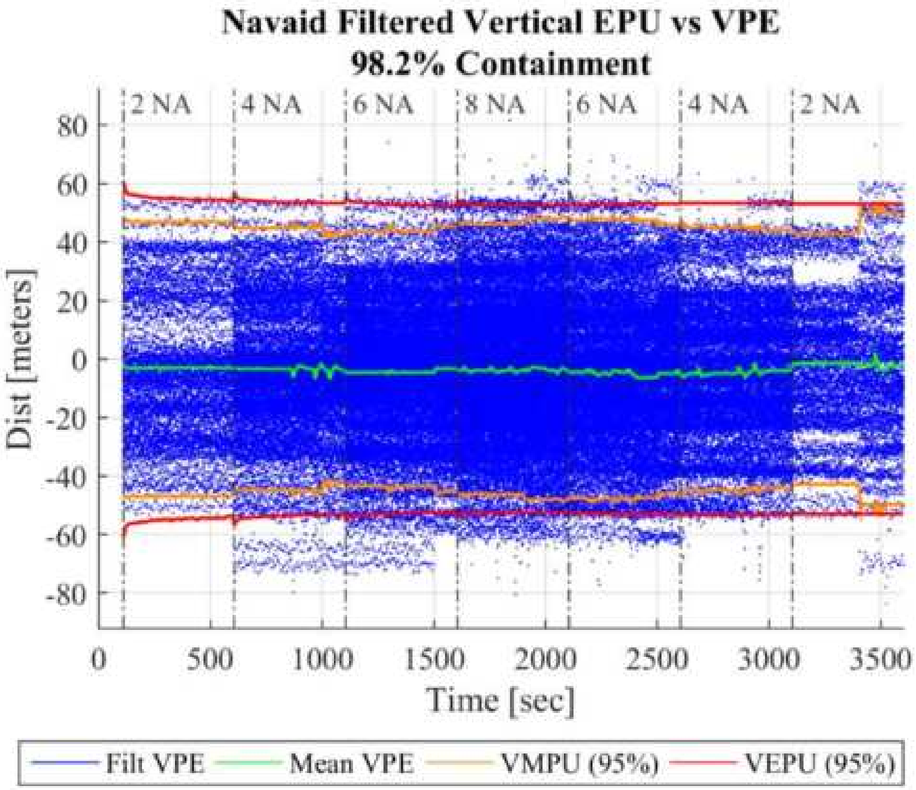

Figure 7-13: Filtered Navigation Aid Vertical Performance for the Varying Number of Navigation Aids Test with Navigation Aid Accuracy Settings of NACp 10 and GVA 2.

A comparison between the mean navigation aid ADS-B reported VPE and the mean navigation aid filtered VPE, along with their associated standard deviations is presented as Figure 7-14. The mean reported VPE was 0.1 meters, while the mean filtered VPE was -3.7 meters, and both exhibited similar standard deviation values. Two noteworthy observations can be taken from these results. First, the results suggests that the filter may be introducing a slight bias in the altitude estimate, although this level of bias is insignificant for this application as this filter was not intended to be deployed in precision navigation applications. Second, the filter appears to be doing little to reject measurement noise in the vertical dimension. This fact is further illustrated in Figure 7-15.

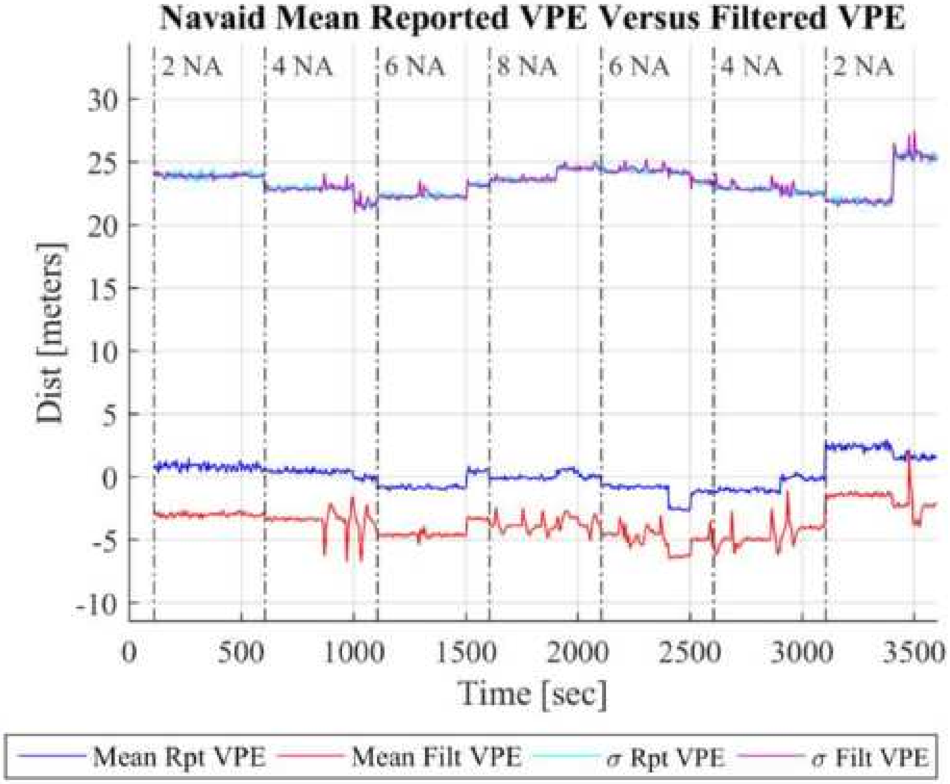

Figure 7-14: Comparison of ADS-B Reported Navigation Aid Vertical Accuracy with Filtered Navigation Aid Lateral Accuracy for the Varying Number of Navigation Aid Test with Navigation Aid Accuracy Settings of NACp 10 and GVA 2.

Figure 7-15 provides a comparison between the ADS-B reported VPE and the filtered VPE. The data appears to suggest that the filter may have been rejecting some measurement noise, but a comparison of the standard deviations reveals that there was little improvement in the filtered VPE. The standard deviation of the reported VPE was 9.9 meters, while the standard deviation of the filtered VPE was 9.7 meters. If a more accurate estimate of navigation aid vertical position were required, then additional filter tuning in the vertical dimensions would be warranted to give more weight to the filter's vertical dimension process model.

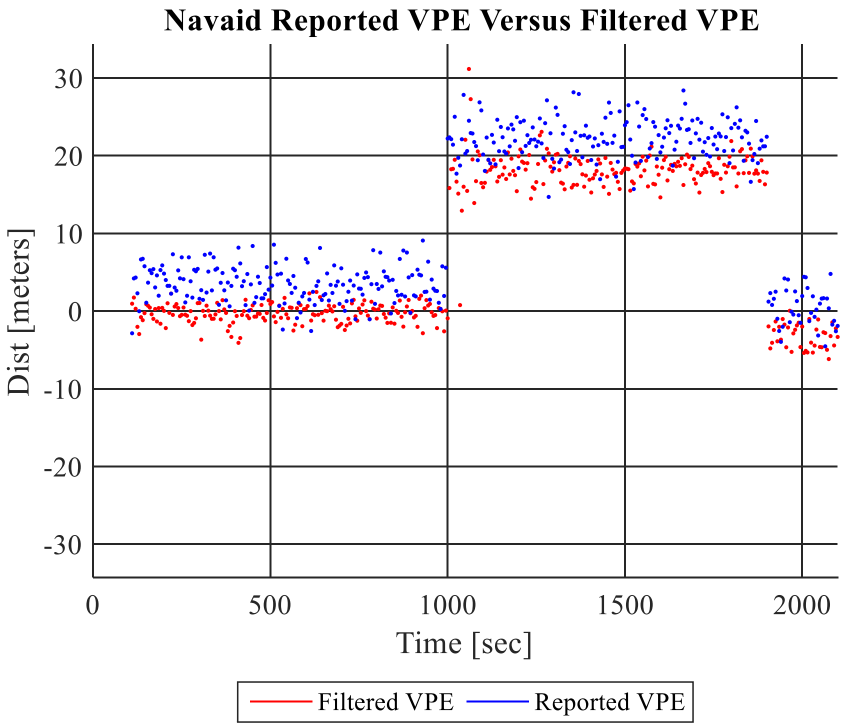

Figure 7-15: Comparison of ADS-B Reported Navigation Aid Vertical Position Error with Filtered Navigation Aid Vertical Position Error for the Varying Number of Navigation Aid Test with Navigation Aid Accuracy Settings of NACp 10 and GVA 2.

7.1.1.7 Summary of Results for the Varying Number of Navigation Aid Test with Navigation Aid Accuracy Settings of NACp 10 and GVA 2

The outcome of this test scenario demonstrated that the algorithm was capable of simultaneously tracking multiple navigation aids while computing a valid estimate of the host vehicle's three-dimensional position. The accuracy of the host vehicle position estimate was a function of the number of navigation aids being tracked, and their geometry relative to the host vehicle. In addition, the results indicated that the algorithm responded well to changing numbers of navigation aids, and produced valid position and accuracy estimates during these transitory periods. This test also showed that in some observable measures, the filter improved the received ADS-B navigation aid position by filtering out measurement noise; thereby, producing a more stable estimate of the navigation aid's position. Finally, this test established that the simulation was providing random inputs that were within the expectation for the NACp and GVA values selected for this experiment.

Table 7-1 provides a set of summary statistics that were collected during this experiment. The same set of statistics was collected for each subsequent test scenario to allow direct comparison of the results using a succinct set of statistics.

Table 7-1: Summary Statistics for the Varying Number of Navigation Aid Test with Navigation Aid Accuracy Settings of NACp 10 and GVA 2.

Summary Statistic	Units	Value	Summary Statistic	Units	Value
Average Host EPU	[m]	1,355	Average Navaid EPU	[m]	88.9
Average Host RPE Mode	[m]	367.2	Average Navaid RPE Mode	[m]	25.5
Host EPU Containment	[%]	97.9	Navaid EPU Containment	[%]	97.8
Average Host VEPU	[m]	466.1	Average Navaid VEPU	[m]	53.4
Average Host VPE	[m]	-4.8	Average Navaid VPE	[m]	-3.7
Host VEPU Containment	[%]	96.7	Navaid VEPU Containment	[%]	98.2

7.1.2 Test Case 2 – NACp 6, GVA 1

This section provides the results observed during Test Case 2 of the Varying Number of Airborne Navigation Aids test. The previous section provided a detailed analysis of many of the filter's characteristics. Several of the same observations could be reiterated in this section; however, the information presented in this section will be limited to observations that are unique to this particular experiment.

7.1.2.1 Filtered Lateral Performance for the Host Vehicle

The host vehicle lateral performance observed during Test Case 2 of the Varying Number of Airborne Navigation Aids test is given in Figure 7-16. The mean EPU was 1,480 meters, with a mean RPE mode of 418 meters. The minimum EPU of 598 meters and minimum RPE mode of 201 meters were again observed when eight navigation aids were available. For this trial, the EPU conservatively contained 97.9% of the RPE values, and was again of similar magnitude to the MPU.

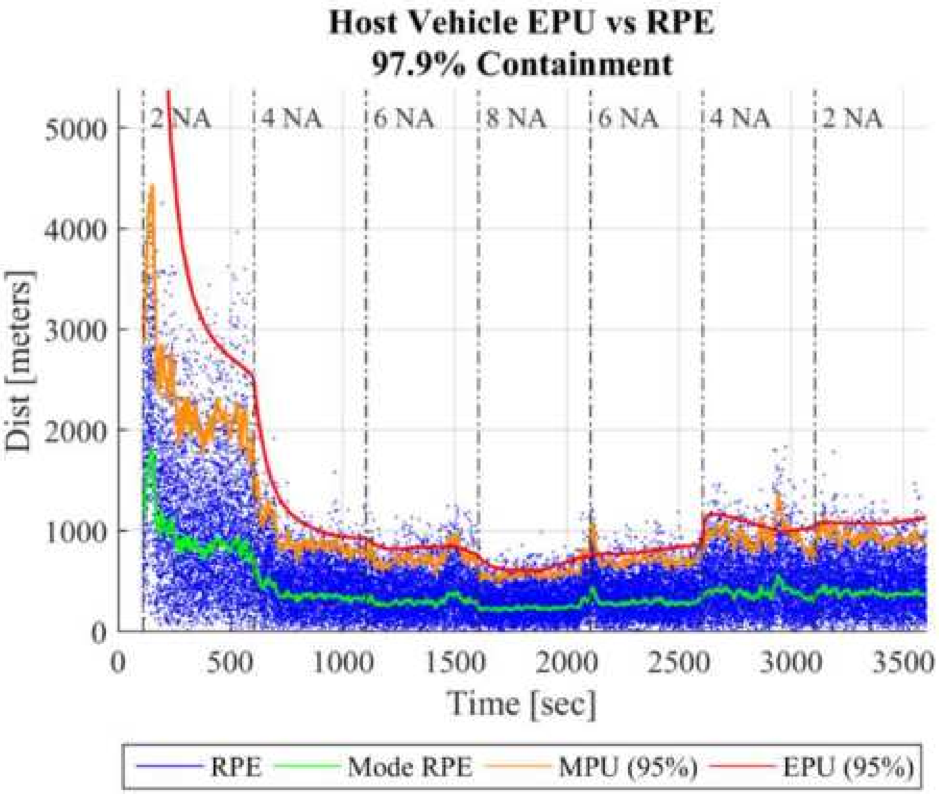

Figure 7-16: Host Vehicle Lateral Performance for the Varying Number of Navigation Aids Test with Navigation Aid Accuracies Setting of NACp 6 and GVA 1.

Compared to Test Case 1, the NACp decreased from 10 to 6, resulting in a navigation aid EPU increase of 5,456% from 10 meters to 555.6 meters (refer to Table 2-6). Despite the large increase in navigation aid EPU, the mean host vehicle EPU only recorded an increase of 9.2% from 1,355 meters to 1,480 meters. Similarly, the mean of the RPE mode increased 13.7% from 367 to 418 meters. This suggests that the accuracy of the host vehicle position estimate may only be weakly correlated with navigation aid accuracy.

7.1.2.2 <u>Characteristics of the Simulated Navigation Aid Lateral ADS-B Data</u>

Per Table 2-6, the NACp level of 6 selected for this test corresponds to a navigation aid EPU of 0.3 NM. Accordingly, the simulation is expected to produce random lateral position reports with a 95% containment level on the order of 556 meters, and an RPE mode on the order of 227 meters. The programmed EPU, instantaneous RPE, RPE mode, and MPU of the lateral ADS-B data produced by the simulation during this test case are provided in Figure 7-17.

The actual MPU shown in Figure 7-17 was 571 meters, versus an expected 556 meters. Likewise, the RPE mode was slightly above the expected 227 meters, coming in at 233 meters. These elevated values are also evident in the fact that the programmed EPU contained only 93.7% of the instantaneous RPE samples. This slight decrease in reported navigation aid containment is again contributed to ADS-B transmission latency. However, due to the decreased navigation aid accuracy settings used for this test case, transmission latency was no longer the dominate factor in the navigation aid RPE. As such, Figure 7-17 validates that the simulation was providing data at the NACp 6 level that was appropriate for this test case.

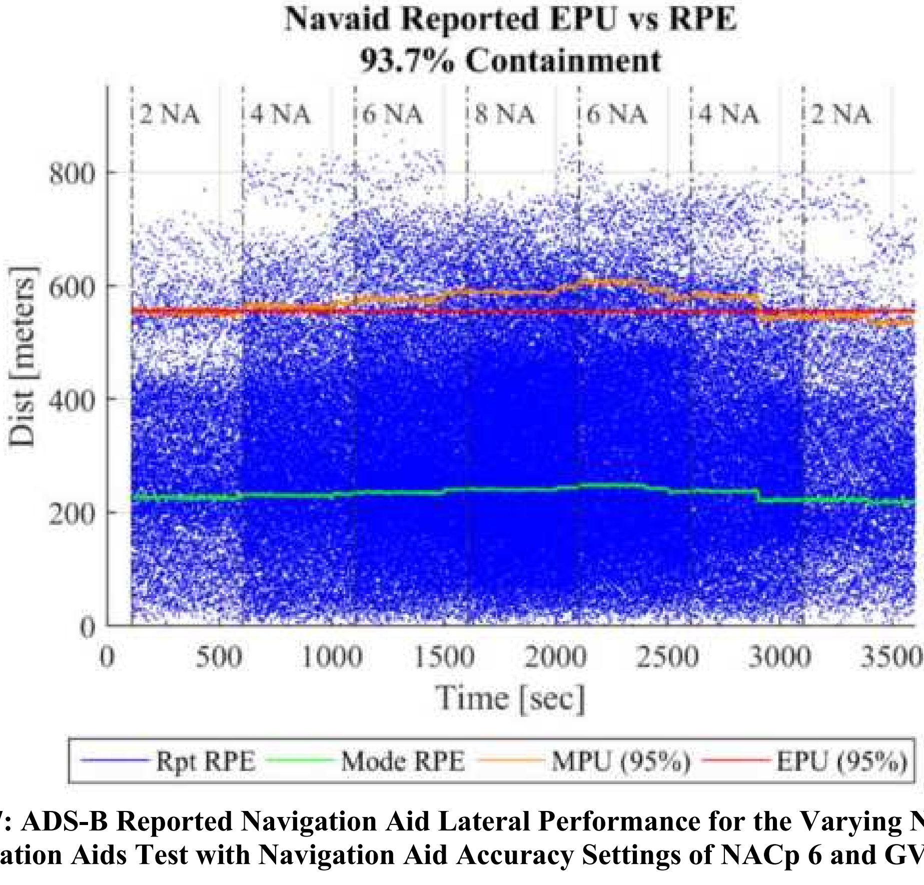

Figure 7-17: ADS-B Reported Navigation Aid Lateral Performance for the Varying Number of Navigation Aids Test with Navigation Aid Accuracy Settings of NACp 6 and GVA 1.

7.1.2.3 *Filtered Lateral Performance for the Tracked Navigation Aids*

The RPE generated from the filter's estimate of the lateral position of the navigation aids is given as Figure 7-18. The average filtered navigation aid RPE mode was 229 meters, with an average EPU value of 593 meters. In this case, the EPU completely contained the MPU. The filter again accounted for the ADS-B transmission latency, resulting in an EPU that contained 96.3% of all position errors.

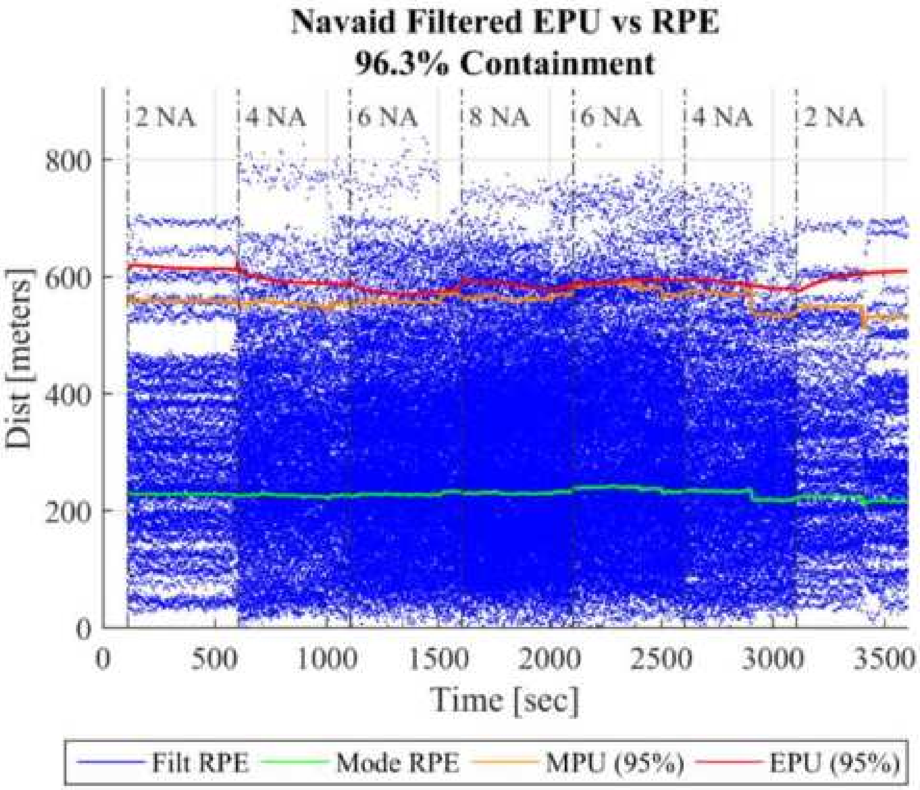

Figure 7-18: Filtered Navigation Aid Lateral Performance for the Varying Number of Navigation Aids Test with Navigation Aid Accuracy Settings of NACp 6 and GVA 1.

A comparison between the mean navigation aid ADS-B reported RPE and the mean navigation aid filtered RPE is provided in Figure 7-19. This figure indicates that the mean filtered RPE decreased below the mean reported RPE level when four or more navigation aids were available. A maximum improvement of 4.5% was achieved when eight navigation aids were included in the filter.

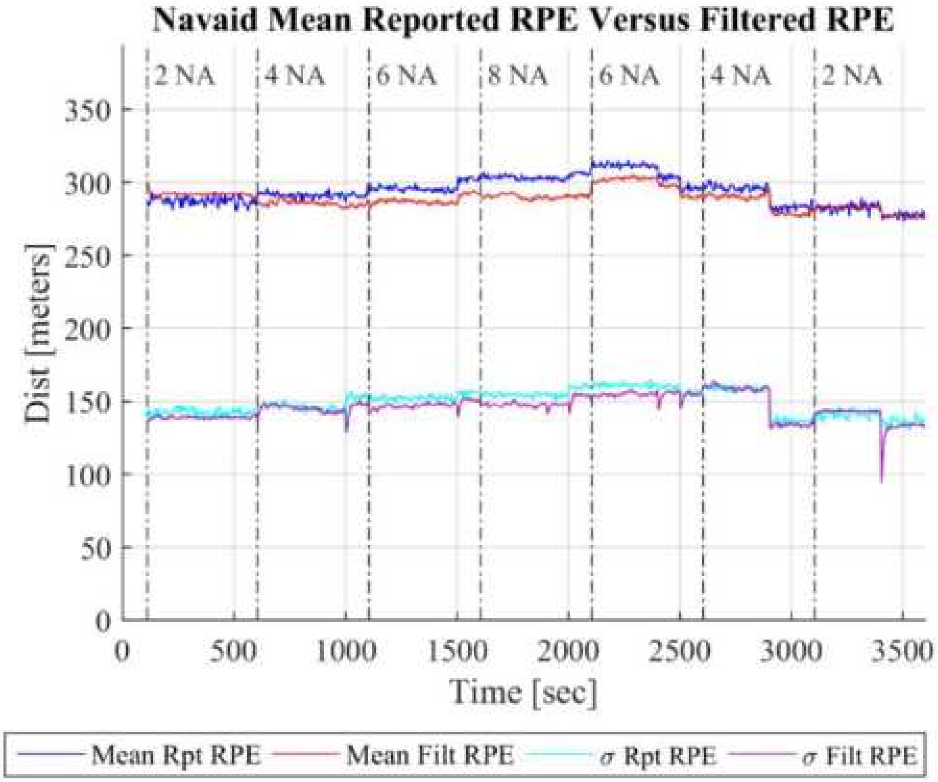

Figure 7-19: Comparison of ADS-B Reported Navigation Aid Lateral Accuracy with Filtered Navigation Aid Lateral Accuracy for the Varying Number of Navigation Aid Test with Navigation Aid Accuracy Settings of NACp 6 and GVA 1.

This improvement in the filtered navigation aid RPE was realized due to the cross correlations established in the filter's covariance matrix. Because the filter tracks all available navigation aids, a correlation is developed between each navigation aid and the host vehicle. Therefore, as the host vehicle's position estimate was improved, the navigation aid position estimates were also improved through the bi-directional AOA observation update process.

7.1.2.4 Filtered Vertical Performance for the Host Vehicle

The vertical performance of the filter observed during Test Case 2 of the Varying Number of Airborne Navigation Aids test is presented as Figure 7-20. The mean VEPU was 473 meters and reflected a 96.3% containment of the VPE. The minimum VEPU of 251 meters was observed when eight navigation aids were incorporated into the filter, while the maximum VEPU of 869 meters occurred during initialization when only two navigation aids were in use.

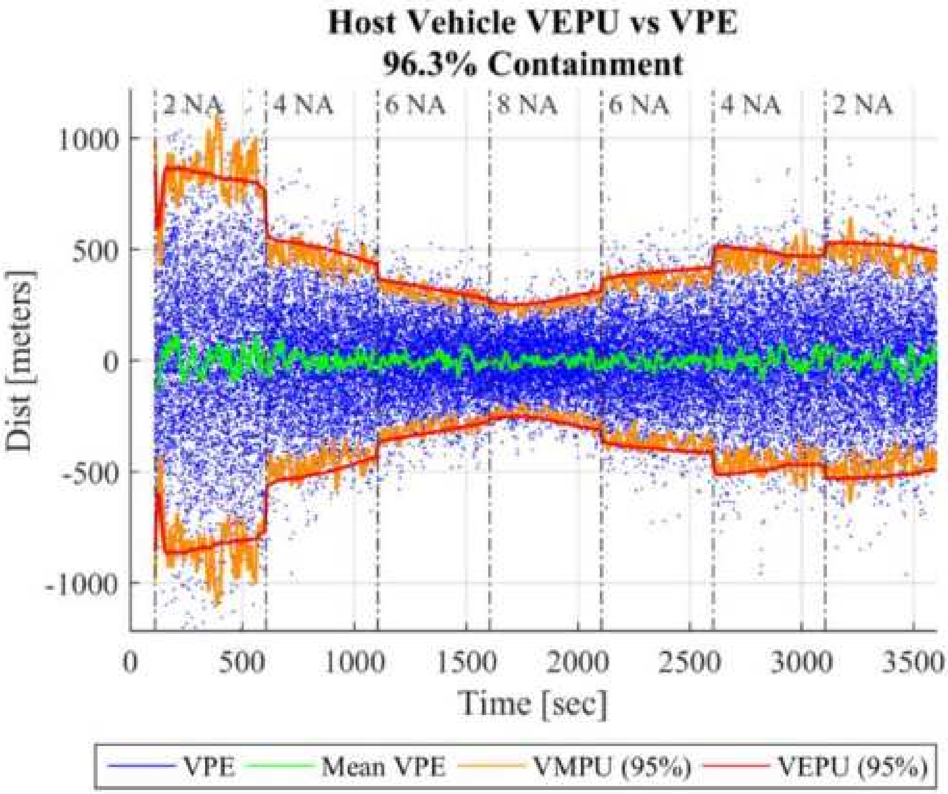

Figure 7-20: Host Vehicle Vertical Performance for the Varying Number of Navigation Aids Tests with Navigation Aid Accuracies Setting of NACp 6 and GVA 1.

For this test case, the GVA was decreased from 2 to 1, which according to Table 2-7 represents a 233% increase in navigation aid VEPU from 45 meters to 150 meters. In response to this increase in navigation aid VEPU, the mean host vehicle VEPU increased from 466 meters to 473 meters, for an increase of 1.57%. This suggests a weak correlation between navigation aid VEPU and the resulting host vehicle VEPU.

7.1.2.5 Characteristics of the Simulated Navigation Aid Vertical ADS-B Data

Figure 7-21 provides the instantaneous VPE, the mean VPE, the programmed VEPU, and the VMPU of the vertical ADS-B data that was generated by the simulation and provided as input to the filter. The plot shows that the VPE was well represented as a zero mean random variable, with a measured 95% containment level (VMPU) of 152 meters. This corresponds well with the GVA 1 containment level of 150 meters that was selected for this test case (refer to

Table 2-7). Finally, the programmed VEPU contained 94.3% of the VPE samples. These observations indicate that the simulation was providing random vertical position reports that correspond to the programmed GVA 1 value.

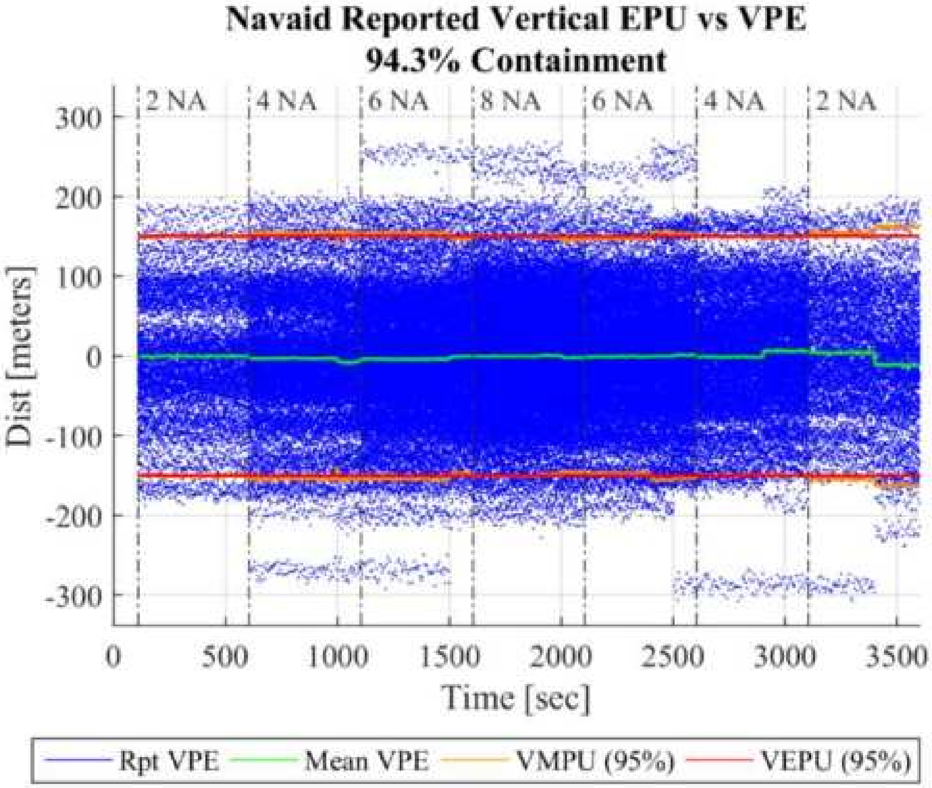

Figure 7-21: ADS-B Reported Navigation Aid Vertical Performance for the Varying Number of Navigation Aids Test with Navigation Aid Accuracy Settings of NACp 6 and GVA 1.

7.1.2.6 Filtered Vertical Performance for the Tracked Navigation Aids

Figure 7-22 provides a graph that demonstrates the filter's performance in tracking the vertical position of the airborne navigation aids. The average VEPU shown in the graph was 155 meters, the average VPE was -5.1 meters, and the average VMPU was 150 meters. These values all correspond with the GVA value of 1 that was selected for this test. These statistics, along with the 95.2% VEPU containment level, indicate that the filter was generating valid vertical position and uncertainty estimates for the navigation aids being tracked.

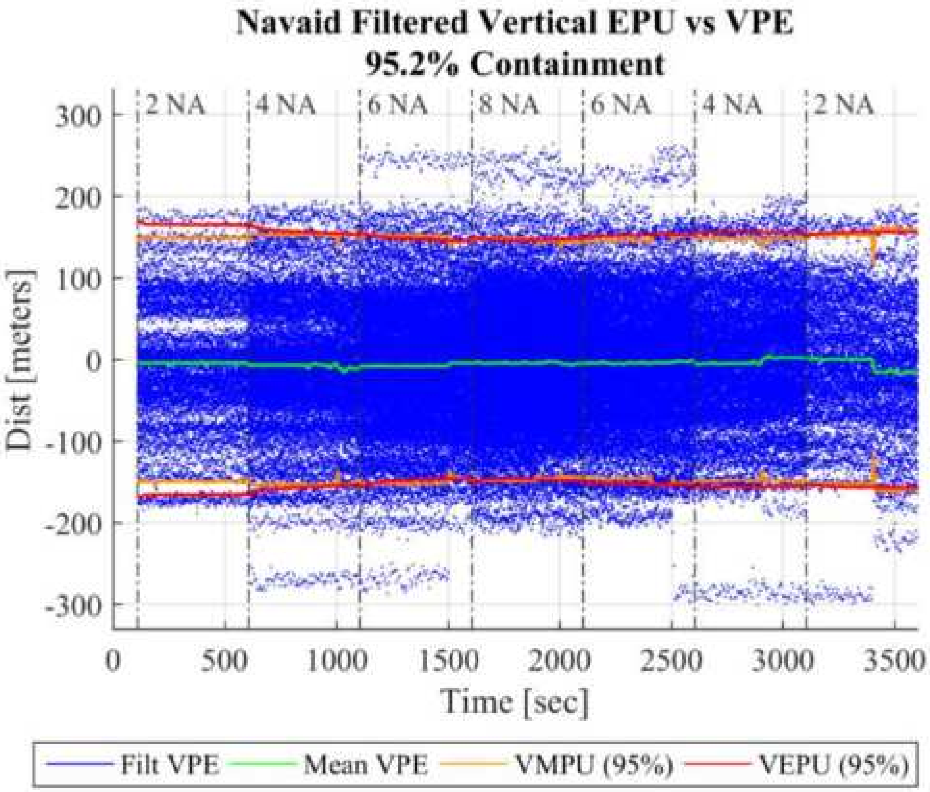

Figure 7-22: Filtered Navigation Aid Vertical Performance for the Varying Number of Navigation Aids Test with Navigation Aid Accuracy Settings of NACp 6 and GVA 1.

Figure 7-23 gives a comparison between the mean navigation aid ADS-B reported VPE and the mean navigation aid filtered VPE. The mean reported VPE was -1.3 meters, while the mean filtered VPE was -5.1 meters. Again the results suggest that the filter may have introduced a small bias in the altitude estimate, and again the filter appears to have done little to reject measurement noise in the vertical dimension.

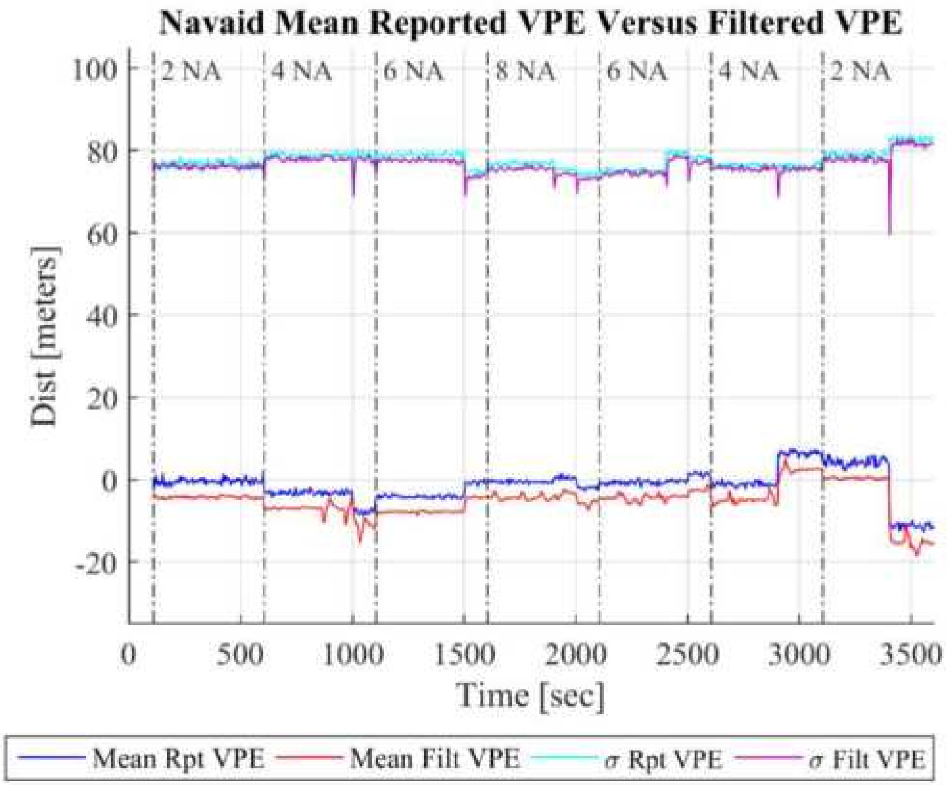

Figure 7-23: Comparison of ADS-B Reported Navigation Aid Vertical Accuracy with Filtered Navigation Aid Lateral Accuracy for the Varying Number of Navigation Aid Test with Navigation Aid Accuracy Settings of NACp 6 and GVA 1.

7.1.2.7 Summary of Results for the Varying Number of Navigation Aid Test with Navigation Aid Accuracy Settings of NACp 6 and GVA 1

The results of this investigation indicate that a decrease in navigation aid accuracy results in a disproportionally small decrease in host vehicle position accuracy. This suggests that the accuracy of the host vehicle position estimate is more dependent on AOA/AOE uncertainty than navigation aid accuracy. This experiment also revealed that in lower NACp cases when four or more navigation aids are available, the filter may be capable of generating a better estimate of the navigation aid's position than reported in the ADS-B data. Table 7-2 presents the summary statistic gathered during this experiment.

Table 7-2: Summary Statistics for the Varying Number of Navigation Aid Test with Navigation Aid Accuracy Settings of NACp 6 and GVA 1.

Summary Statistic	Units	Value	Summary Statistic	Units	Value
Average Host EPU	[m]	1,480	Average Navaid EPU	[m]	592.5
Average Host RPE Mode	[m]	417.5	Average Navaid RPE Mode	[m]	228.6
Host EPU Containment	[%]	97.9	Navaid EPU Containment	[%]	96.3
Average Host VEPU	[m]	473.4	Average Navaid VEPU	[m]	154.6
Average Host VPE	[m]	-1.6	Average Navaid VPE	[m]	-5.1
Host VEPU Containment	[%]	96.3	Navaid VEPU Containment	[%]	95.2

7.1.3 Test Case 3 – NACp 2, GVA 1

The present section provides an overview of the observations from Test Case 3 of the Varying Number of Airborne Navigation Aids test. As with the previous section, the observations provided here will be limited to those that are unique to this particular test case.

7.1.3.1 Filtered Lateral Performance for the Host Vehicle

The lateral performance of the host vehicle's filtered position as observed during Test Case 3 of the Varying Number of Airborne Navigation Aids test is provided in Figure 7-24. The mean EPU was 6,839 meters, with a mean RPE mode of 2,454 meters. The minimum EPU and RPE again occurred when eight navigation aids were being tracked. In the case, the minimum EPU and RPE were 3,333 meters and 1,176 meters respectively. Finally, A 95.7% EPU containment rate was observed for this trial.

Compared to Test Case 1, the NACp decreased from 10 to 2, resulting in a navigation aid EPU increase from 10 meters to 7,408 meters (refer to Table 2-6) for a 73,980% increase in navigation aid EPU. The enormous increase in navigation aid EPU led to a relatively moderate 405% increase in the mean host vehicle EPU, and a similarly moderate 568% increase in the mean of the RPE mode. This is further evidence that the host vehicle position accuracy may be only weakly correlated with navigation aid accuracy.

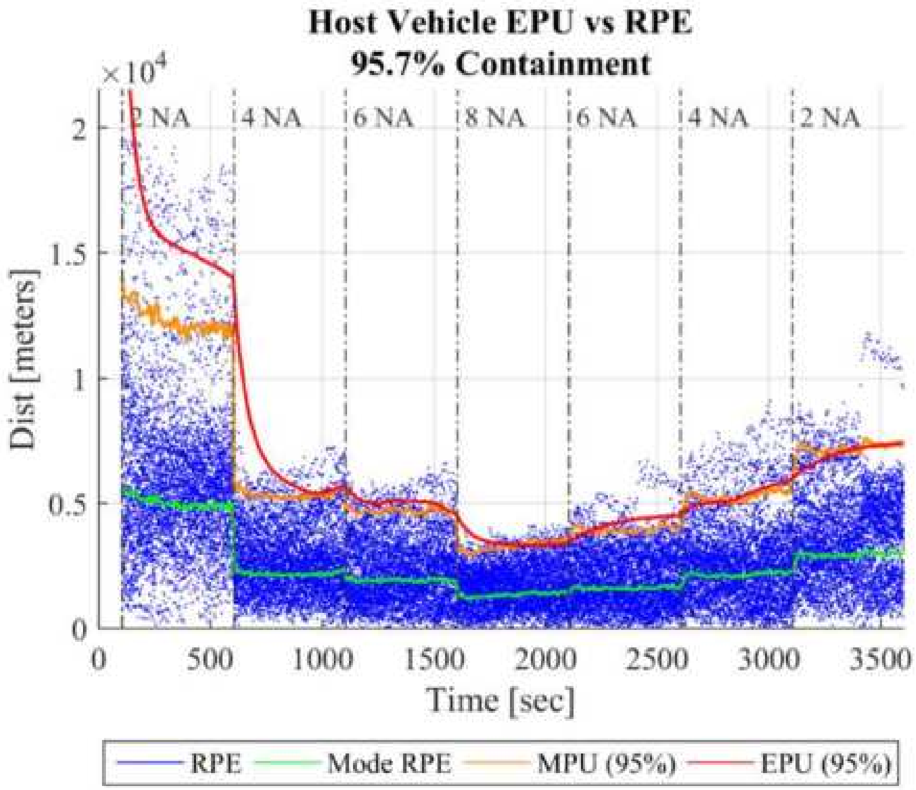

Figure 7-24: Host Vehicle Lateral Performance for the Varying Number of Navigation Aids Test with Navigation Aid Accuracies Setting of NACp 2 and GVA 1.

7.1.3.2 Characteristics of the Simulated Navigation Aid Lateral ADS-B Data

In accordance with Table 2-6, the NACp level of 2 selected for this test corresponds to a navigation aid EPU of 4.0 NM. Accordingly, the expectation is that the simulation should produce random lateral position reports with a 95% containment level on the order of 7,408 meters and an RPE mode on the order of 3,026 meters. To validate this, the programmed EPU, instantaneous RPE, RPE mode, and MPU of the lateral ADS-B data produced by the simulation during this test case is provided in Figure 7-25.

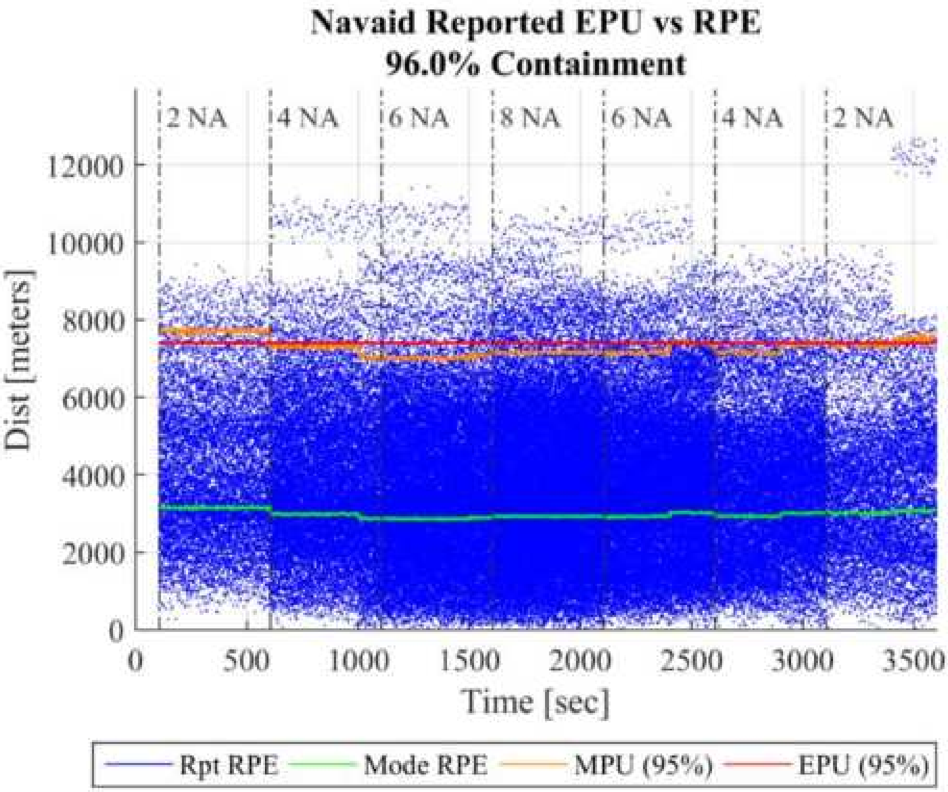

Figure 7-25: ADS-B Reported Navigation Aid Lateral Performance for the Varying Number of Navigation Aids Test with Navigation Aid Accuracy Settings of NACp 2 and GVA 1.

The average MPU shown in Figure 7-25 was 7,297 meters, versus an expected 7,408 meters. The RPE mode was also slightly below the expected 3,026 meters, coming in at 2,981 meters. For this test case, these slight variations are not attributed to ADS-B transmission latency because the observed values are now below the levels configured for the test case. That is, latency would be expected to increase the error rather than decrease it. The fact that the simulated RPE statistics fall below the configured levels is also evident in the fact that the configured EPU contained 96% of the simulated RPE samples instead of the desired 95%. These slight variations represent an error of approximately 1.5% and are considered acceptable due to the finite sample size used to generate the statistics. As such, Figure 7-25 confirms that the simulation is providing data at the NACp 2 level that is appropriate for this test case.

7.1.3.3 Filtered Lateral Performance for the Tracked Navigation Aids

The navigation aid RPE generated from the filter's estimate of the lateral position of the navigation aids is given as Figure 7-26. The average filtered navigation aid RPE mode was 2,617 meters, with an average EPU value of 6,847 meters. Except for the pair of two navigation aid intervals, the EPU completely contained the MPU. In total, the EPU generated by the filter contained 96.1% of all navigation aid lateral position errors.

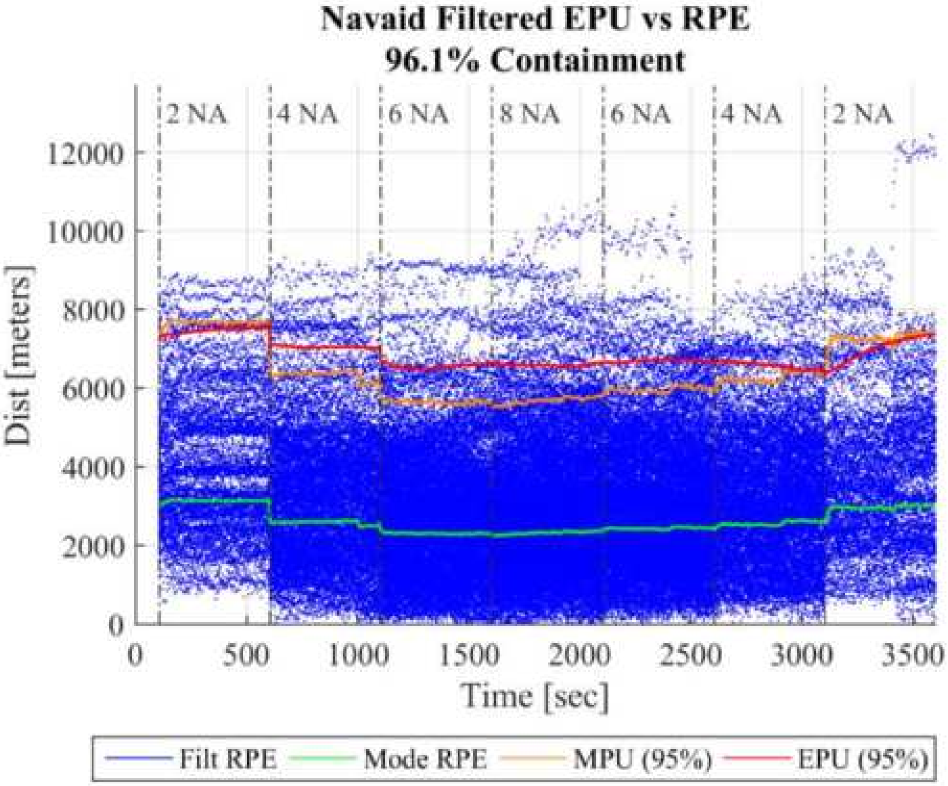

Figure 7-26: Filtered Navigation Aid Lateral Performance for the Varying Number of Navigation Aids Test with Navigation Aid Accuracy Settings of NACp 2 and GVA 1.

Figure 7-27 presents the comparison between the mean navigation aid ADS-B reported RPE and the mean navigation aid filtered RPE. Again, due to cross correlations, the mean filtered RPE decreased below the mean reported RPE level when four or more navigation aids were available. An EPU improvement up to 33.3% and a corresponding RPE improvement of up to 16.1% were realized when eight navigation aids were available. These observations confirm that the bi-directional AOA observation update process can generate a better navigation aid position estimate than what is received in the ADS-B data set, particularly for inaccurate navigation aids.

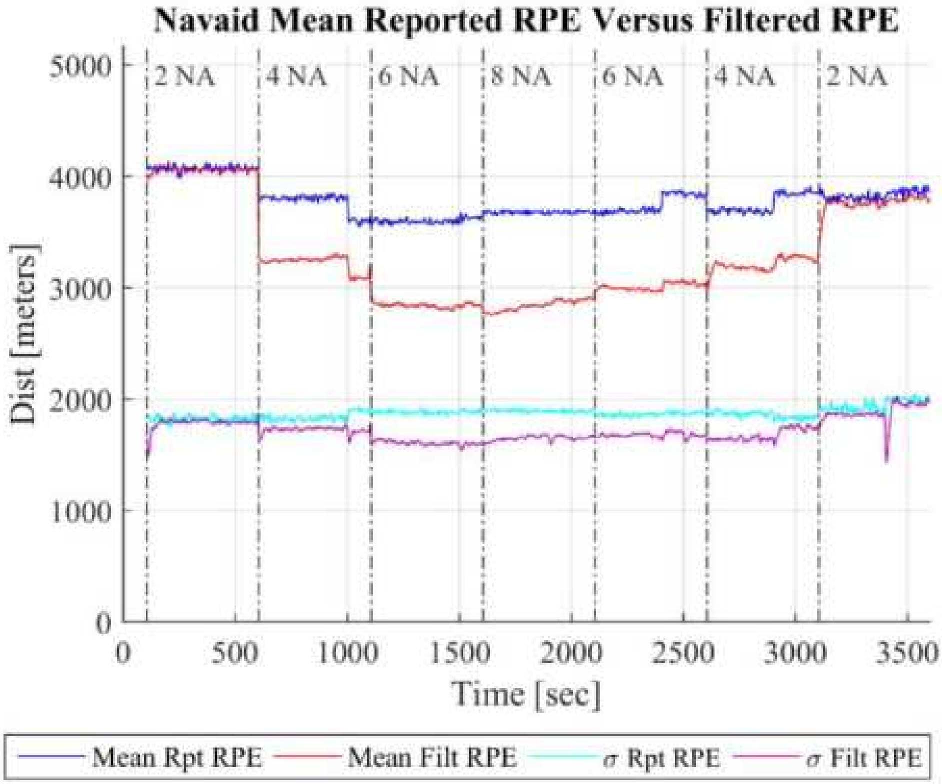

Figure 7-27: Comparison of ADS-B Reported Navigation Aid Lateral Accuracy with Filtered Navigation Aid Lateral Accuracy for the Varying Number of Navigation Aid Test with Navigation Aid Accuracy Settings of NACp 2 and GVA 1

7.1.3.4 Filtered Vertical Performance for the Host Vehicle

The host vehicle vertical performance of the filter observed during Test Case 2 of the Varying Number of Airborne Navigation Aids test is shown in Figure 7-28. The mean VEPU was 497 meters and reflected a 96.3% containment of the VPE. The minimum VEPU of 262 meters was observed when eight navigation aids were incorporated into the filter; while the maximum VEPU of 958 meters occurred during initialization when only two navigation aids were in use.

The GVA value remained at 1 for this test case, yet the mean VEPU experienced a 4.9% increase over the result observed during Test Case 2 that also used a GVA value of 1. Likewise, the minimum VEPU exhibited a 4.3% increase. This increase in VEPU is attributed to the corresponding decrease in navigation aid lateral position accuracy that occurred between Test Case 2 and Test Case 3. As evident in equation (4-39), the estimated AOE used to develop the measurement update innovations is a function of the distance between the host vehicle and the airborne navigation aid. As such, errors in the navigation aid lateral position couple into the host vehicle vertical position estimate, resulting in a decrease in the host vehicle vertical position accuracy.

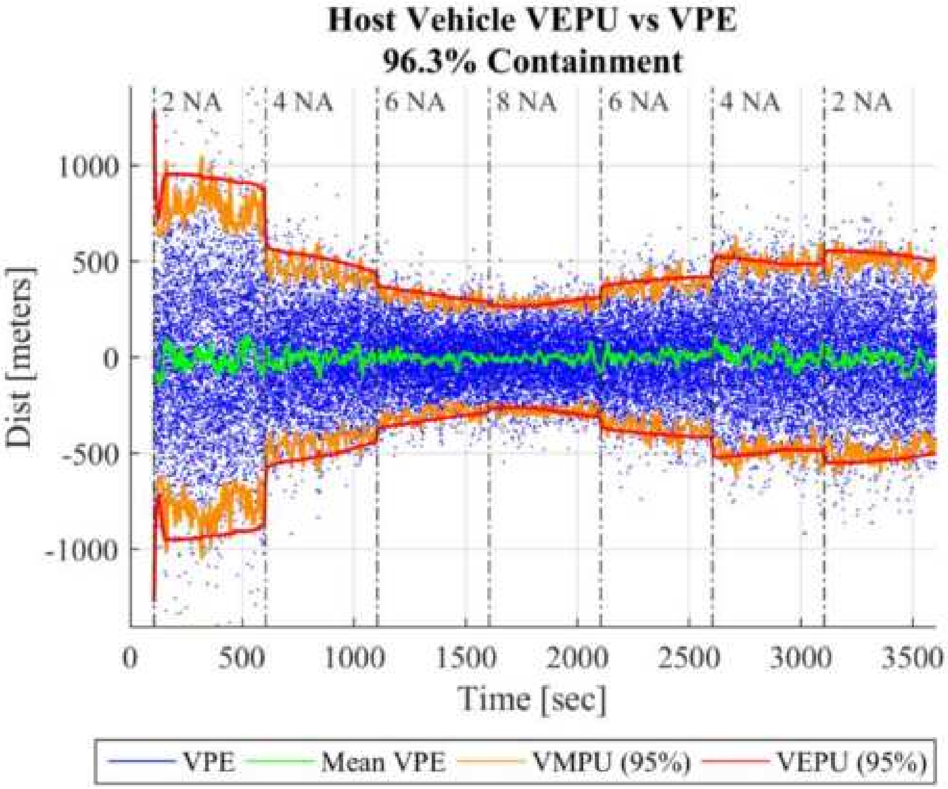

Figure 7-28: Host Vehicle Vertical Performance for the Varying Number of Navigation Aids Tests with Navigation Aid Accuracies Setting of NACp 2 and GVA 1.

7.1.3.5 Characteristics of the Simulated Navigation Aid Vertical ADS-B Data

Figure 7-29 provides the instantaneous VPE, the mean VPE, the programmed VEPU, and the VMPU of the vertical ADS-B data that was generated by the simulation and provided as input to the filter. This plot is quite similar to Figure 7-21 because both test case utilize a GVA value of 1. For the present test case, the plot shows that the VPE was well represented as a zero mean random variable, with a measured 95% containment level (VMPU) of 145 meters. This corresponds well with the GVA 1 containment level of 150 meters that was selected for this test case (refer to Table 2-7). Finally, the programmed VEPU contained 95.8% of the VPE samples. These observations indicate that the simulation was providing random vertical position reports that correspond with the GVA programmed for this test.

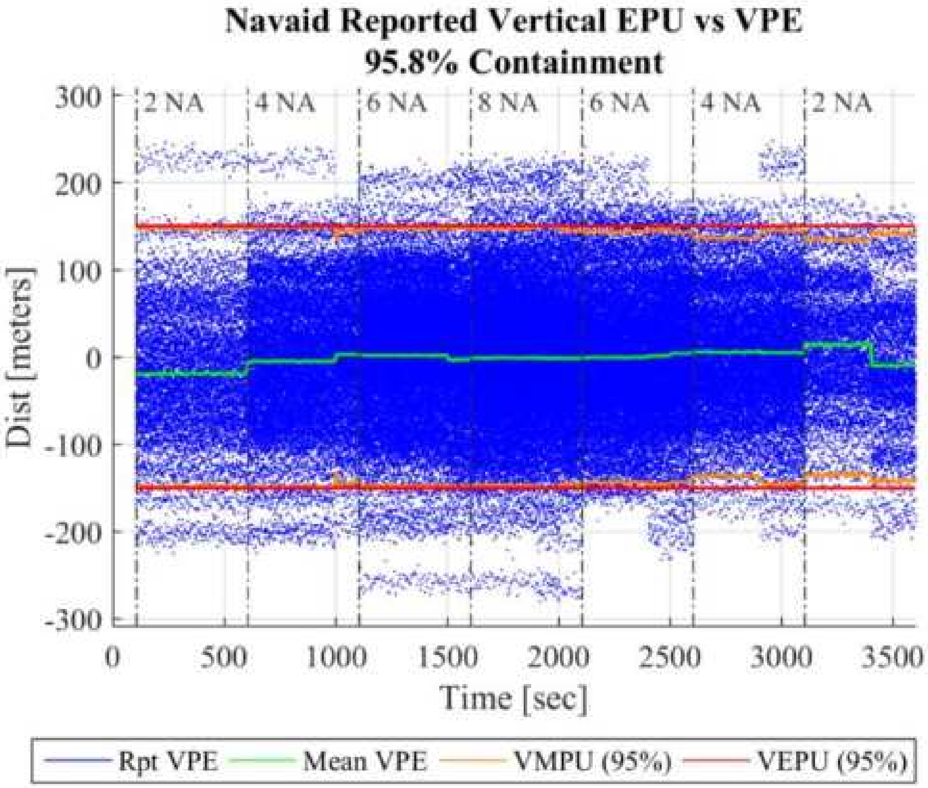

Figure 7-29: ADS-B Reported Navigation Aid Vertical Performance for the Varying Number of Navigation Aids Test with Navigation Aid Accuracy Settings of NACp 2 and GVA 1.

7.1.3.6 Filtered Vertical Performance for the Tracked Navigation Aids

A plot that shows the filter's performance in tracking the vertical position of the airborne navigation aids is given as Figure 7-30. The average VEPU shown in the graph was 160 meters, the average VPE was -5.3 meters, and the average VMPU was 143 meters. The VEPU completely contained the VMPU for the duration of the plotted data. These values all correspond with the GVA value of 1 that was selected for this test. These statistics, along with the 96.9% VEPU containment level, indicate that the filter generated valid vertical position and uncertainty estimates for the navigation aids being tracked.

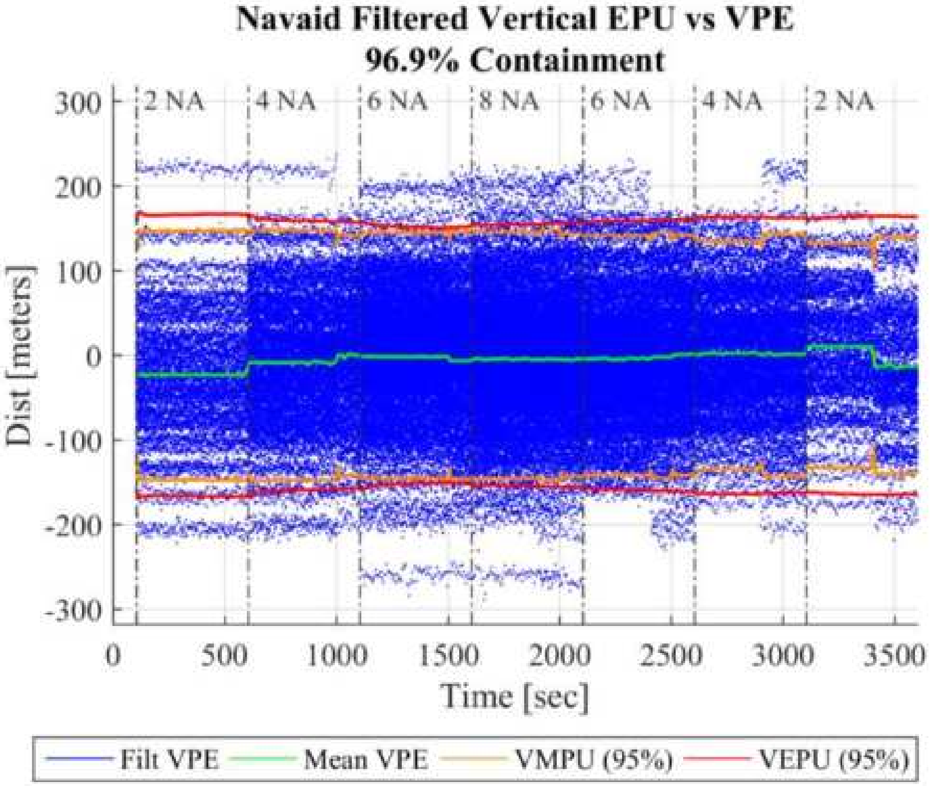

Figure 7-30: Filtered Navigation Aid Vertical Performance for the Varying Number of Navigation Aids Test with Navigation Aid Accuracy Settings of NACp 2 and GVA 1.

The comparison between the mean navigation aid ADS-B reported VPE and the mean navigation aid filtered VPE is provided in Figure 7-31. The mean reported VPE was -1.5 meters, while the mean filtered VPE was -5.3 meters. Again the results suggest that the filter may be introducing a small bias in the altitude estimate and the filter still appears to be doing little to reject measurement noise in the vertical dimension.

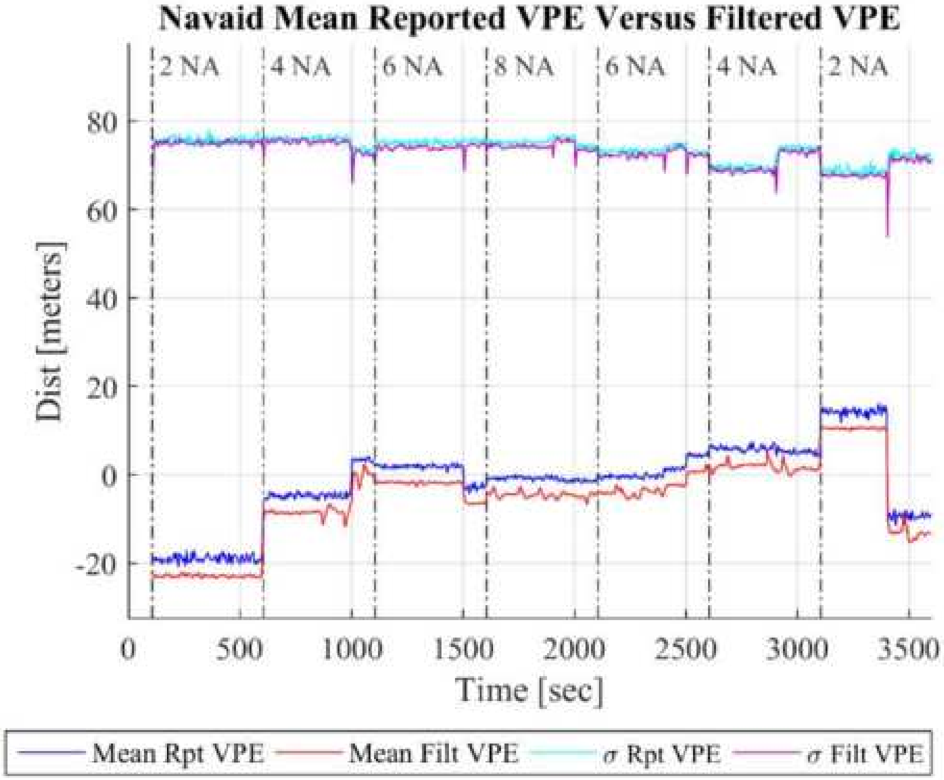

Figure 7-31: Comparison of ADS-B Reported Navigation Aid Vertical Accuracy with Filtered Navigation Aid Lateral Accuracy for the Varying Number of Navigation Aid Test with Navigation Aid Accuracy Settings of NACp 2 and GVA 1.

7.1.3.7 Summary of Results for the Varying Number of Navigation Aid Test with Navigation Aid Accuracy Settings of NACp 2 and GVA 1

The information gleamed from this test case provided further evidence that the accuracy of the host vehicle position estimate is only weakly correlated with airborne navigation aid accuracy. An increase in navigation aid EPU of 73,980% resulted in a corresponding 405% increase in host vehicle EPU. This test case also provided evidence that navigation aid lateral position errors can affect the both the host vehicle's lateral and vertical position estimates. Finally, it was confirmed that for low accuracy navigation aids, the filter can indeed generate a better estimate of the navigation aid's true position than what was being reported in the ADS-B data. Table 7-3 presents the summary statistic gathered during this experiment.

Table 7-3: Summary Statistics for the Varying Number of Navigation Aid Test with Navigation Aid Accuracy Settings of NACp 2 and GVA 1.

Summary Statistic	Units	Value	Summary Statistic	Units	Value
Average Host EPU	[m]	6,839	Average Navaid EPU	[m]	6,847
Average Host RPE Mode	[m]	2,454	Average Navaid RPE Mode	[m]	2,617
Host EPU Containment	[%]	95.7	Navaid EPU Containment	[%]	96.1
Average Host VEPU	[m]	496.7	Average Navaid VEPU	[m]	159.9
Average Host VPE	[m]	-6.6	Average Navaid VPE	[m]	-5.3
Host VEPU Containment	[%]	96.3	Navaid VEPU Containment	[%]	96.9

7.2 Random Navigation Aid Test Results

The purpose of this section is to provide the results of the Random Navigation Aid Test Scenario described in Section 6.2. Due to the random nature of this experiment, the test was only executed once.

7.2.1 Filtered Lateral Performance for the Host Vehicle

The host vehicle lateral performance observed during the Random Navigation Aid Test is given in Figure 7-32. For this test scenario, the simulation was configured to provide ten valid navigation aids, all having random accuracy and random flight profiles. These navigation aids were configured to be valid from the beginning of each run, and to remain valid throughout. Because the expectation is that the ten navigation aids were available to the filter throughout each run, the graph does not denote the number of navigation aids in use.

The minimum observed EPU value was 393 meters with a mean of 580 meters. The minimum RPE mode was 111 meters, and the mean RPE mode was 162 meters. For this test, the EPU value represented a 97.4% containment level and contained the MPU in steady state. These observations indicate that the filter's host vehicle lateral position estimate was more accurate with ten random navigation aids than it was during all of the Varying Number of Navigation Aids tests whose results are detailed in Section 7.1. This indicates that the filter's lateral performance continued to improve as the number of navigation aids in view increased beyond eight.

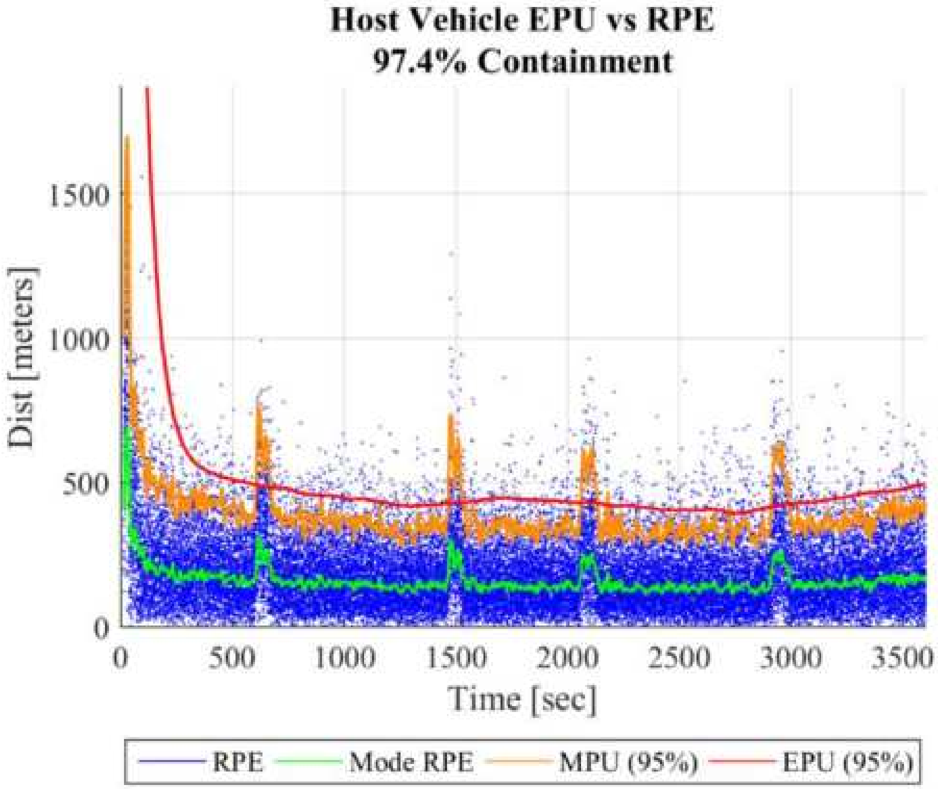

Figure 7-32: Host Vehicle Lateral Performance for the Random Navigation Aids Test.

The RPE mode and the MPU exhibited divergence from the mean when the host vehicle experienced a turn. This implies that the filter followed the process model more than the measurements during these transitory periods. For this non-precision navigation application, these errors are considered acceptable because the mode of the RPE was still well contained by the EPU, and relatively few instantaneous RPE samples exceeded the EPU. It may be possible to limit the position errors experienced during host vehicle maneuvering by implementing an adaptive filter design that adjusts the process noise during state changes. However, some method of detecting these state changes would be required, and this may include integration of additional sensors. As such, this is considered beyond of the scope of the current research.

7.2.2 *Characteristics of the Simulated Navigation Aid Lateral ADS-B Data*

Figure 7-33 provides the instantaneous RPE, the Rayleigh mode of the RPE, the programed EPU, and the MPU of the simulated lateral ADS-B data that was provided as input to the filter. It is obvious from the plot that the MPU greatly exceeds the average of the randomly selected EPU values configured for the navigation aids. This is expected for high accuracy navigation aids due to the ADS-B transmit latency as described in Section 7.1.1.2. The takeaway from this graph is that the simulation was providing random ADS-B position reports that appear reasonable given the range of possible NACp values selected for this test.

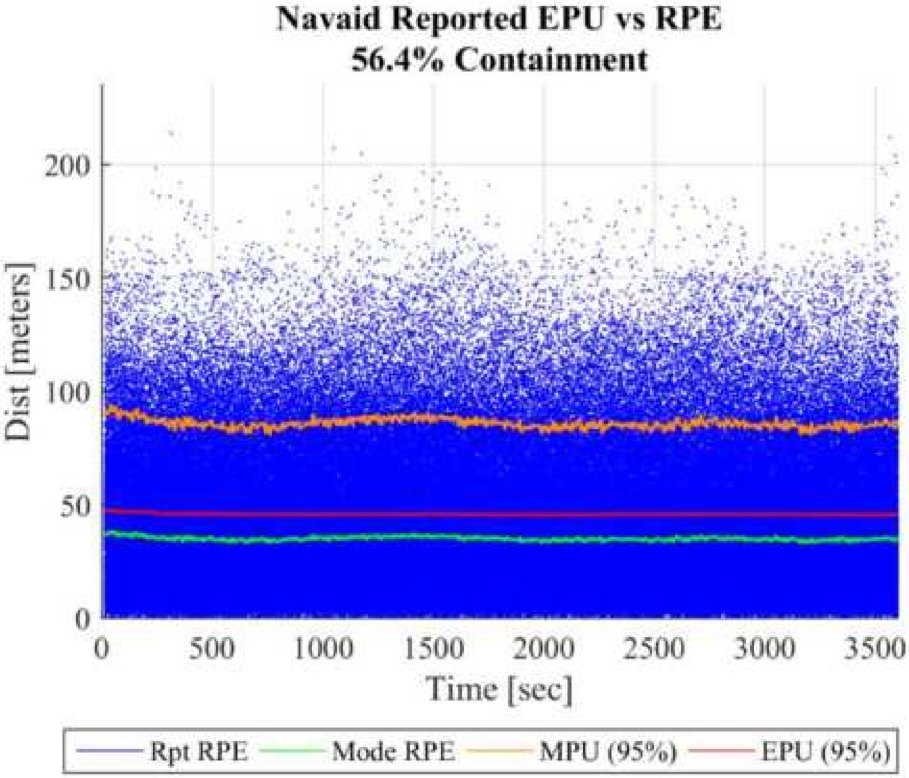

Figure 7-33: ADS-B Reported Navigation Aid Lateral Performance for the Random Navigation Aids Test.

7.2.3 *Filtered Lateral Performance for the Tracked Navigation Aids*

The RPE generated from the filter’s estimate of the lateral position of the navigation aids is given in Figure 7-34. The average filtered navigation aid RPE mode was 39.4 meters, with an average EPU value of 111 meters. In this test case, the EPU showed convergence to the MPU level and contained 96.4% of all position errors.

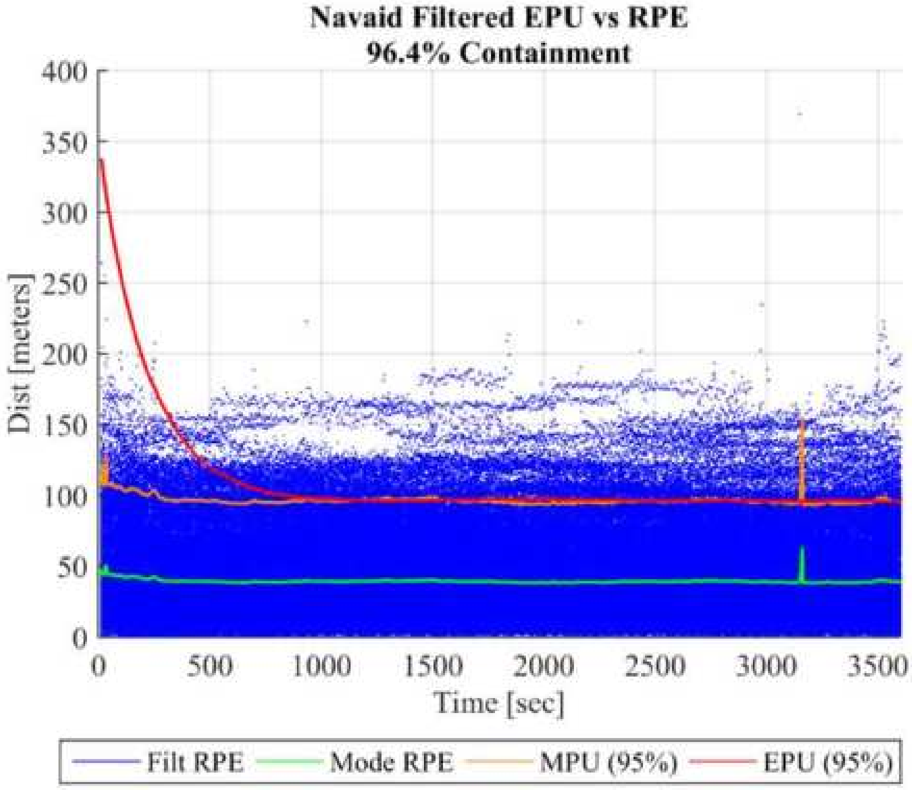

Figure 7-34: Filtered Navigation Aid Lateral Performance for the Random Navigation Aids Test.

The RPE mode and MPU at time stamp 3,150 seconds of Figure 7-34 show a discontinuity. Analysis of the data revealed a brief period of RPE outliers for a single navigation aid during one of the 50 runs. The instantaneous RPE for a single navigation aid increased to 369 meters at time 3,150 seconds, then increased to 886 meters at time 3,155 seconds, and peaked at 1,601 meters at time 3,160 seconds. Due to the selection of 400 meters as the maximum Y-axis value, the RPE step to 369 meters is the only outlier visible in the figure. Because this outlier only occurred for a single navigation aid in the midst of a 50-run ensemble, it is impossible to ascertain the exact cause. The most likely causes would be navigation aid position relative to the host vehicle, excessive AOA white noise generated by the simulation, repeated ADS-B data collisions, or other temporary loss of ADS-B data from the navigation aid that persisted for more than 20 seconds. The fact that the divergence was relatively short duration, and self-corrected, suggests that this was not an issue with a singularity or numeric precision of the filter, but rather a transient event that led to a temporary divergence in the filter's estimate this one navigation aid's position.

7.2.4 Filtered Vertical Performance for the Host Vehicle

The performance of the filter's host vehicle vertical position estimation as observed during the Random Navigation Aid test is given as Figure 7-35. The mean VEPU was 205 meters and reflected a 95.9% containment of the VPE. The minimum observed VEPU was 176 meters, and the maximum observed VEPU was 261 meters. These observations indicate that the filter's host vehicle vertical position estimate was more accurate with ten random navigation aids than it was during all of the Varying Number of Navigation Aids tests whose results are detailed in Section 7.1. This shows that the filter's vertical performance continued to improve as the number of navigation aids in view increased beyond eight.

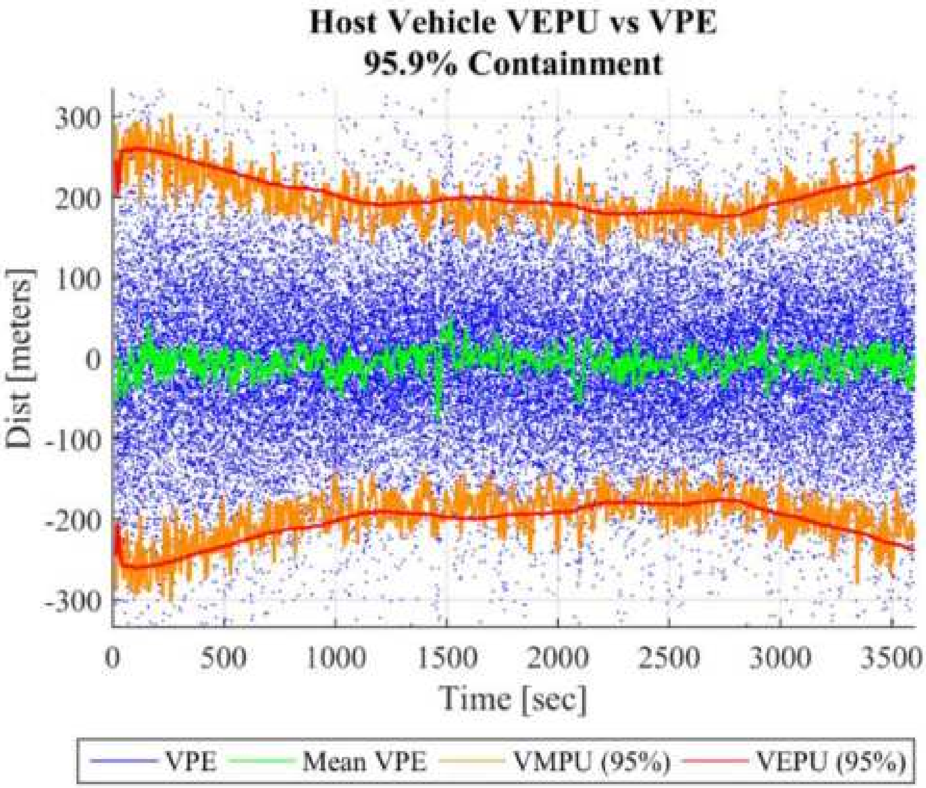

Figure 7-35: Host Vehicle Vertical Performance for the Random Navigation Aids Tests.

7.2.5 Characteristics of the Simulated Navigation Aid Vertical ADS-B Data

Figure 7-36 provides the instantaneous VPE, the mean VPE, the programmed VEPU, and the VMPU of the simulated vertical ADS-B data that was provided as input to the filter. This plot shows that the simulation was providing random vertical inputs to the algorithm, as denoted by the blue scatter points representing the VPE. The data also indicates that the VPE was well represented as a zero mean random variable, with a measured 95% containment level (VMPU) of 116 meters versus an average randomly selected EPU of 103 meters. The randomly selected EPU contained 95.2% of the VPE samples. These results confirm that the simulation was providing valid inputs for this test case.

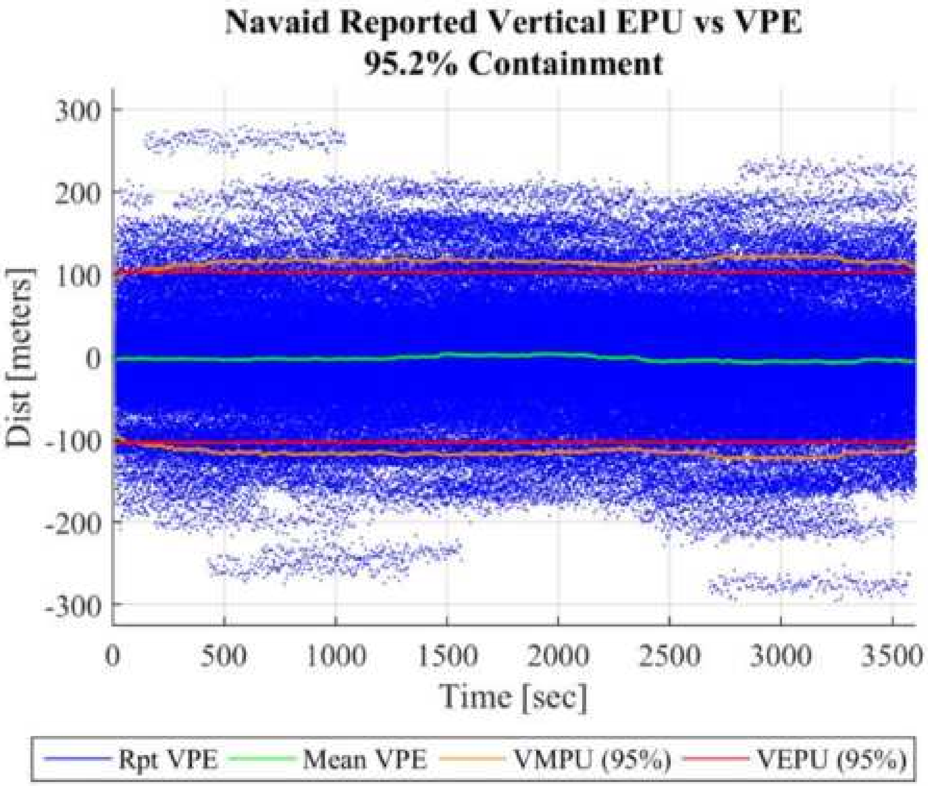

Figure 7-36: ADS-B Reported Navigation Aid Vertical Performance for the Random Navigation Aids Test.

7.2.6 Filtered Vertical Performance for the Tracked Navigation Aids

A plot demonstrating the filter's performance in tracking the vertical position of the airborne navigation aids is given as Figure 7-37. The average VEPU was 101 meters, the average VPE was -5.7 meters, and the average VMPU was 113 meters. The filtered VEPU contained 95.3% of the instantaneous VPE and was of similar magnitude to the MPU. This suggests that the filter generated valid vertical position and accuracy estimates for the tracked navigation aids.

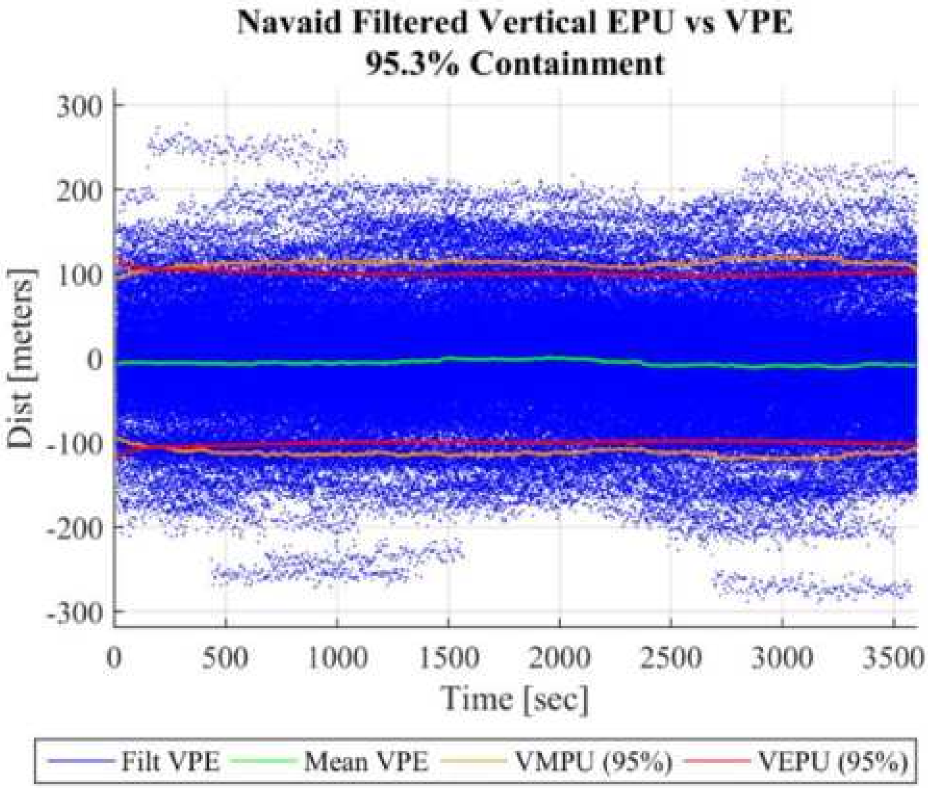

Figure 7-37: Filtered Navigation Aid Vertical Performance for the Random Navigation Aids Test.

7.2.6.1 Summary of Results for the Random Navigation Aid Test Scenario

This test case was selected to provide a set of inputs that would more closely resemble a real world scenario, where the both the accuracy and flight paths of the airborne navigations would be random. With the exception of a single navigation aid lateral position divergence during one instance of the 50-run ensemble, the algorithm performed well. In fact, both the lateral and vertical host vehicle accuracy exceeded the best accuracy observed during the Variable Number of Airborne Navigation Aids testing of Section 7.1. This suggests that the accuracy of the host vehicle position estimates can be expected to improve as the number of airborne navigation aids increases beyond eight. Table 7-4 provides the set of summary statistics that were collected during this experiment.

Table 7-4: Summary Statistics for the Random Navigation Aid Test.

Summary Statistic	Units	Value	Summary Statistic	Units	Value
Average Host EPU	[m]	579.7	Average Navaid EPU	[m]	110.5
Average Host RPE Mode	[m]	162.1	Average Navaid RPE Mode	[m]	39.4
Host EPU Containment	[%]	97.4	Navaid EPU Containment	[%]	96.4
Average Host VEPU	[m]	205.0	Average Navaid VEPU	[m]	100.6
Average Host VPE	[m]	-5.7	Average Navaid VPE	[m]	-5.7
Host VEPU Containment	[%]	95.9	Navaid VEPU Containment	[%]	95.3

7.3 Parametric Test Results

This section presents the observations made during the Parametric Test event described in Section 6.3. As discussed in Section 6.3, a detailed analysis of the results is not provided. Instead, the same set of summary statistics gathered for the previous tests was collected for the parametric tests to allow for a direct comparison of the results. Supporting plots similar to those presented in Section 7.1 and 7.2 are available in APPENDIX J. To avoid presentation of a significant amount of duplicate data, the summary statistics for these parametric tests are provided in Table 7-7 and Table 7-8 of Section 7.4.

7.3.1 Filtered Lateral Performance for the Host Vehicle

Figure 7-38 presents a graphical summary of the host vehicle lateral accuracy as observed during parametric testing. For this plot, the host vehicle accuracy is graphed as a function of NACp value, with a different trace for each of the different number of navigation aids selected as explained in Table 6-3 of Section 6.3. Figure 7-38A shows the mean host vehicle EPU as a function of NACp value, while Figure 7-38B shows the mean host vehicle RPE mode as a function of NACp. These plots collectively illustrate that there was relatively little improvement of host vehicle lateral accuracy as the navigation aids NACp values improved from 6 to 10. However, a significant improvement in host vehicle EPU is evident as the navigation aid NACp improved from 2 to 6. Table 7-5 quantifies the level of navigation aid EPU improvement, and corresponding host vehicle EPU improvement, for each line segment in the figures. This table is interpreted as the percent decrease in EPU or RPE for a given step change in NACp value. For example, row 1 of Table 7-5 indicates that when eight navigation aids were available, a NACp increase from 2 to 6 resulted in a 92.5% decrease in the navigation aid EPU, a 78.7% decrease in the host vehicle EPU, and an 81.5% decrease in the host vehicle RPE mode.

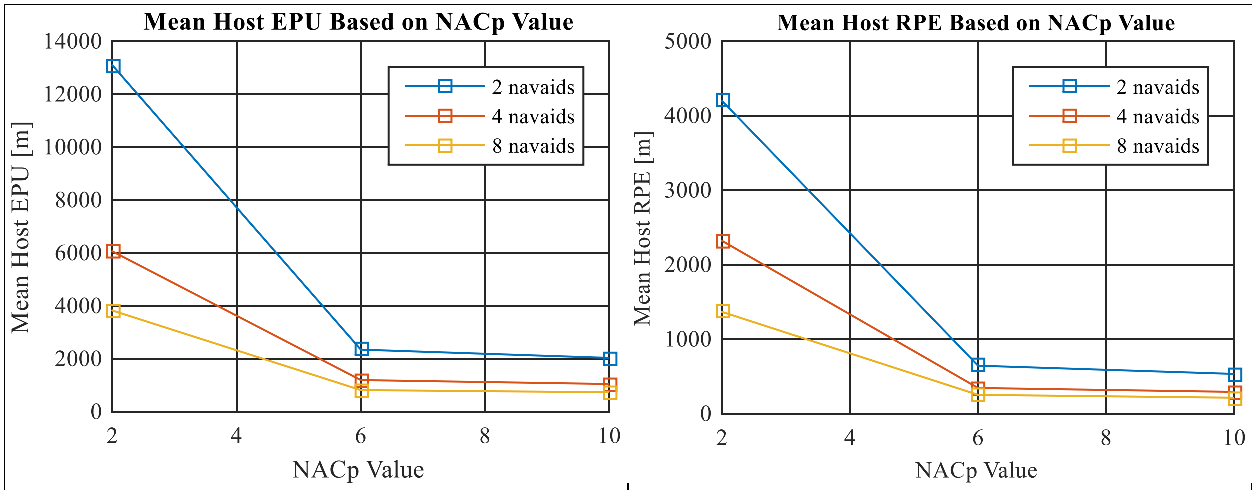

Figure 7-38A: Average Host Vehicle EPU. **Figure 7-38B: Average Host Vehcile RPE.**

Figure 7-38: Parametric Testing – Average Host Vehicle Lateral Accuracy as a Function of NACp.

Table 7-5: Percent Change of EPU and RPE Improvement as a Function of Increasing NACp.

Number Navaids	Beginning NACp	Ending NACp	Navigation Aid EPU Change	Host Vehicle EPU Change	Host Vehicle RPE Change
8	2	6	-92.5%	-78.7%	-81.5%
8	6	10	-98.2%	-10.0%	-15.3%
4	2	6	-92.5%	-80.3%	-85.2%
4	6	10	-98.2%	-12.4%	-15.2%
2	2	6	-92.5%	-82.1%	-84.6%
2	6	10	-98.2%	-13.3%	-17.2%

Analysis of Table 7-5 reveals that a NACp step from 6 to 10 constitutes a larger percentage decrease in navigation aid EPU than a NACp step from 2 to 6. However, the host vehicle EPU experiences a larger percent decrease when the navigation aid NACp stepped from 2 to 6. This occurs because when high quality (high NACp) navigation aids are used, the dominant error source in the host vehicle position estimation process is the AOA/AOE measurement uncertainty. As the NACp of the navigation aids falls below 6, the uncertainty in the navigation aid position begins to contribute more significantly to the host vehicle's position uncertainty.

With respect to Figure 7-38, it is interesting to note that a more accurate host vehicle position was observed using just 2 high quality navigation aids than was achieved using 8 low quality navigation aids. This suggests that the number of available navigation aids may only marginally contribute to the overall accuracy. In an effort to confirm this theory, Figure 7-39 presents the same data as Figure 7-38, but the accuracy is now plotted as a function of the number of available navigation aids for each of the three NACp levels.

Figure 7-39 collaborates Figure 7-38 and shows that there was minimal difference in host vehicle accuracy between NACp 6 and NACp 10 over the range of available navigation aids. Figure 7-39 also demonstrates that the host vehicle lateral accuracy improved for all NACp levels as the number of available navigation aids increased, although the level of improvement showed diminishing returns for high quality navigation aids. Increasing the number of available navigation aids contributed most significantly for NACp 2 level navigation aids. Table 7-6 quantifies the level of improvement as a percentage decrease in the host vehicle EPU and RPE values as the number of navigation aids increased.

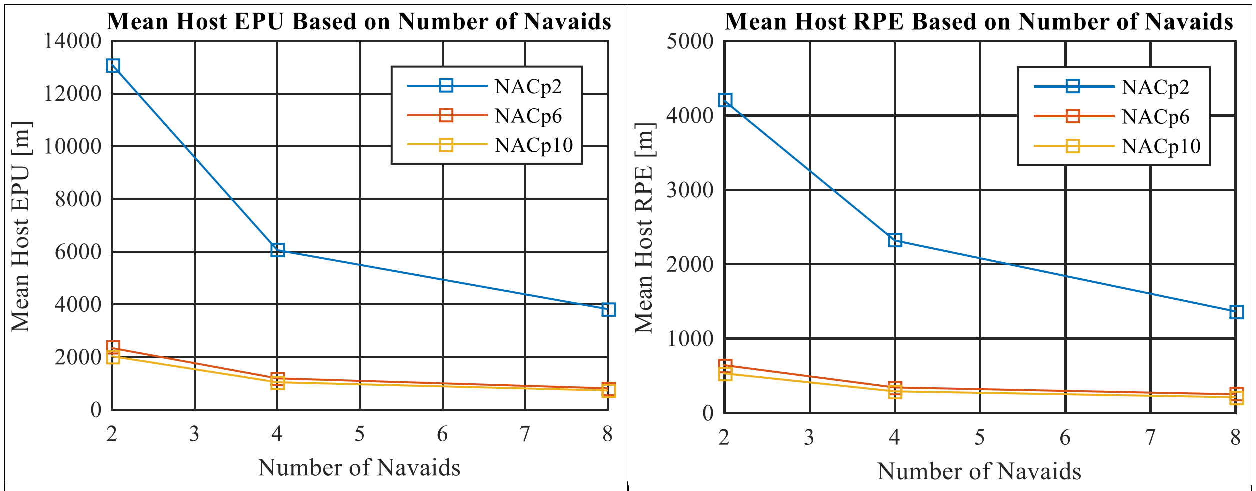

Figure 7-39A: Average Host Vehicle EPU. **Figure 7-39B: Average Host Vehicle RPE.**

Figure 7-39: Parametric Testing – Average Host Vehicle Lateral Accuracy as a Function of Number of Navigation Aids.

Table 7-6: Percent Change of EPU and RPE as a Function of Number of Navigation Aids.

NACp	Beginning Number of Navigation Aids	Ending Number of Navigation Aids	Host Vehicle EPU Change	Host Vehicle RPE Change
10	2	4	-48.5%	-45.2%
10	4	8	-29.9%	-26.9%
6	2	4	-49.1%	-46.5%
6	4	8	-31.7%	-26.7%
2	2	4	-53.7%	-44.7%
2	4	8	-36.9%	-41.1%

7.3.2 Filtered Lateral Performance for the Tracked Navigation Aids

A summary plot of the filter's estimate of the tracked navigation aid lateral accuracy observed during parametric testing is given in Figure 7-40. For this plot, the navigation aid accuracy is graphed as a function of NACp value, with a different trace for each of the different number of navigation aids evaluated. Figure 7-40A provides the mean filtered navigation aid EPU as a function of NACp, while Figure 7-40B gives the mean filtered navigation aid RPE as a function of NACp. These plots show that for high quality navigation aids, the number of navigation aids being tracked has little bearing on the accuracy of navigation aid position estimate. However, in the NACp 2 case, a discernable improvement in both navigation aid EPU and RPE are evident. This reaffirms similar observations made in Test Case 3 of the Varying Number of Navigation Aids tests and described in Section 7.1.1.3.

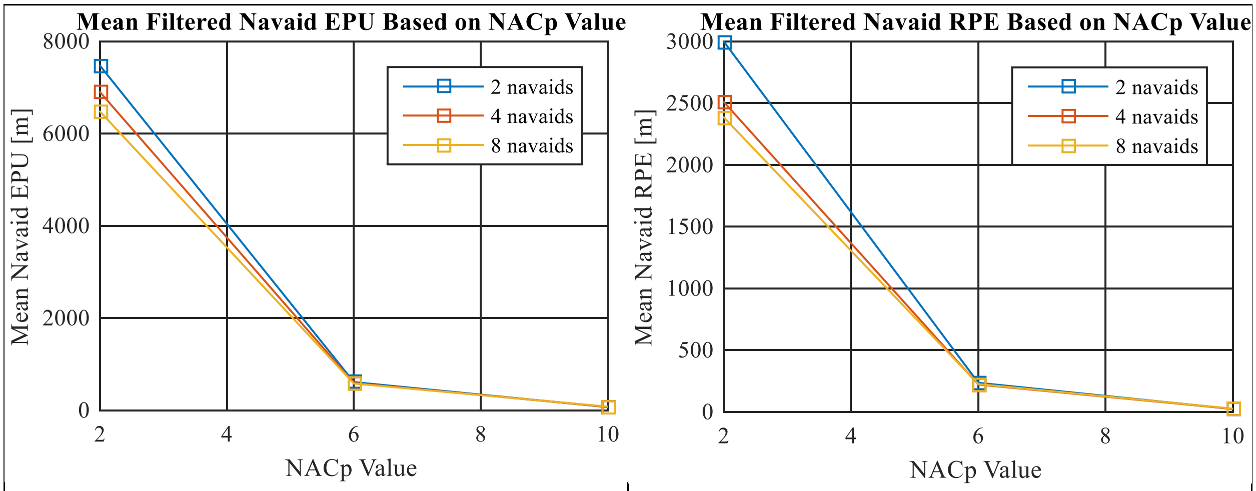

Figure 7-40A: Mean Filtered Navaigation Aid EPU. **Figure 7-40B: Mean Filtered Navaigation Aid RPE.**

Figure 7-40: Parametric Testing – Mean Filtered Navigation Aid Lateral Accuracy as a Function of NACp Value.

As shown in Table 7-7, the mean filtered navigation aid EPU for two NACp 2 navigation aids was 7,470 meters and the mean RPE mode was 2,999 meters. Similarly, the mean filtered navigation aid EPU for eight NACp 2 navigation aids was 6,476 meters with a mean RPE mode of 2,385 meters. This constitutes a 13.3% decrease in navigation aid EPU and a 20.5% decrease in navigation aid RPE. This improvement is realized due to a combination of the cross correlations established in the filter's covariance matrix and the relative accuracy of the AOA measurements as compared to the navigation aid's position uncertainty.

7.3.3 Filtered Vertical Performance for the Host Vehicle

Figure 7-41 provides a summary plot of the host vehicle vertical position accuracy as noted during parametric testing. Figure 7-41A presents the mean host vehicle VEPU as a function of NACp value, while Figure 7-41B provides the mean host vehicle VEPU as a function of the number of navigation aids being tracked. Plots of the mean VPE have not been included because the VPE is one-dimensional measurement that was essentially zero-mean. The mean VPE numerical values are provided in Table 7-8.

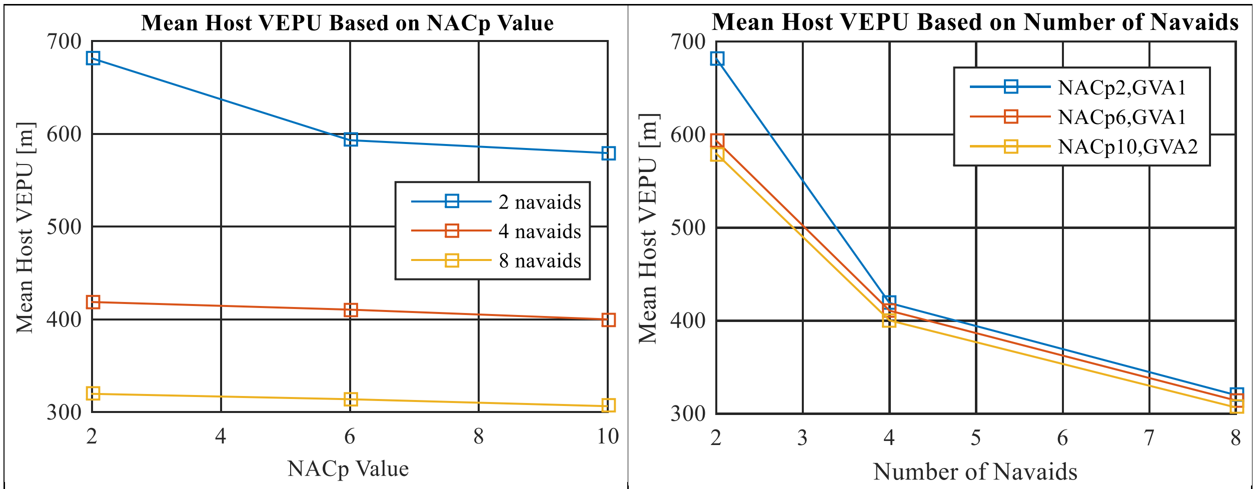

Figure 7-41A: Average Host Vehcile VEPU as a function of NACp Value.

Figure 7-41B: Average Host Vehcile VEPU as a function of Number of Navigation Aids.

Figure 7-41: Parametric Testing – Average Host Vehicle Vertical Accuracy.

Figure 7-41A reveals that a minor improvement in host vehicle vertical position accuracy was realized as the navigation aid lateral accuracy (NACp) improved. In the two navigation aid case, the mean VEPU decreased 15.0% as the NACp increased from 2 to 10. For the four and eight navigation aid cases, the mean VEPU decreased 4.5% and 4.2% respectively. This relatively minor improvement in host vehicle vertical accuracy as the navigation aid lateral accuracy improved is due to the fact that the host vehicle vertical position measurement update process is a function of the distance from the host vehicle to the navigation aid and their associated position uncertainties. As such, improvements in the navigation aid lateral accuracy result in a horizontal coupling to improve vertical accuracy as well.

Figure 7-41B shows that the mean host vehicle VEPU decreased as the number of available navigation aids increased. The greatest improvement occurred on the transition from two to four navigation aids, with an average VEPU improvement of 33.4%. On the other hand, the VEPU improvement as the number of available navigation aids increased from four to eight was 23.5%. In both cases, the number of available navigation aids doubled, but the resulting accuracy improvement displayed diminishing returns.

7.3.4 Filtered Vertical Performance for the Tracked Navigation Aids

The filtered estimate of the tracked navigation aid vertical accuracy during parametric testing is provided in Figure 7-42. Again, plots of the mean VPE have not been included because the VPE is one-dimensional measurement that was essentially zero-mean. The mean VPE numerical values are provided in Table 7-8.

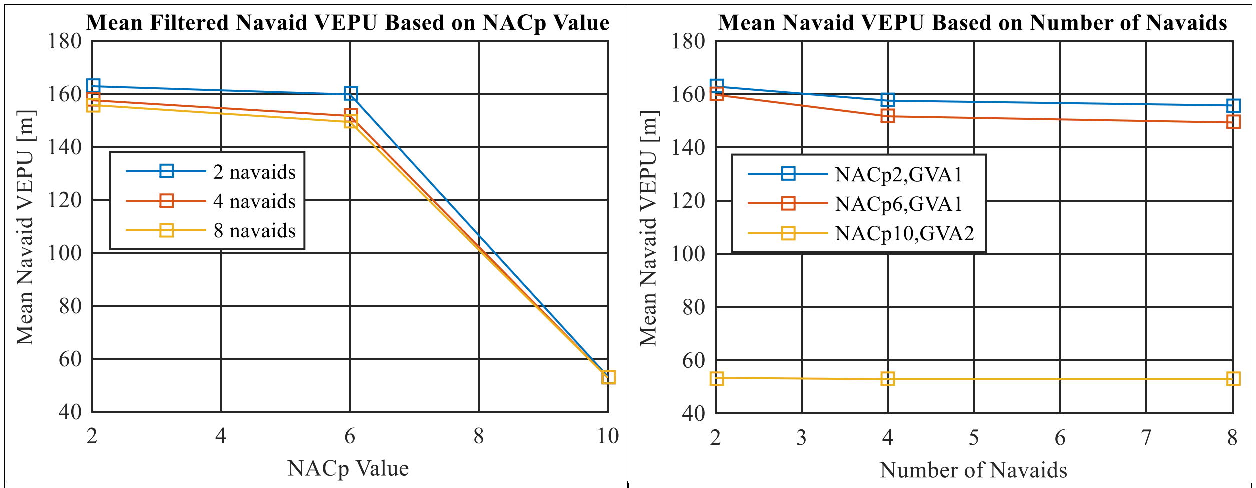

Figure 7-42A: Average Filtered Navigation Aid VEPU as a function of NACp Value.

Figure 7-42B: Average Filtered Navigation Aid VEPU as a function of Number of Navigation Aids.

Figure 7-42: Parametric Testing – Average Filtered Navigation Aid Vertical Accuracy.

Figure 7-42A shows a slight improvement in the navigation aid filtered VEPU as the navigation aid lateral accuracy improved from NACp 2 to NACp 6. This was again due to the vertical measurement updates being a function of navigation aid lateral position. The plot indicates a significant improvement in navigation aid VEPU as the NACp value increased from 6 to 10; however, the NACp 10 test scenario utilized a GVA of 2, while the NACp 6 test utilized a GVA of 1. This VEPU accuracy improvement was a function of improved reported navigation aid vertical accuracy rather than a function of filtering and improving NACp.

Figure 7-42B indicates that the number of available navigation aids bears little consequence to the filtered navigation aid VEPU. For the NACp 10, GVA 2 case, a miniscule 0.9% improvement was observed as the number of navigation aids increased from two to eight. For the NACp 2 and NACp 6 cases, with the GVA set to 1, the average navigation aid filtered VEPU improvement was 5.4%. Per Table 2-7, the expected vertical accuracy for GVA 1 is on the order of 150 meters, and the expected vertical accuracy for GVA 2 is 45 meters. Both plots of Figure 7-42 indicate mean VEPU values of these magnitudes.

7.3.5 Effects of AOA/AOE Uncertainty on Host Vehicle Position Accuracy

The purpose of this section is to provide a comparison of the filter's performance given an AOA/AOE uncertainty of 6.0°, as reported by Faragher in Section 3.1.2, as opposed to the 0.7° AOA/AOE uncertainty achieved by Reck is Section 3.1.1. For this experiment, the eight navigation aid, NACp 10 parametric test scenario was repeated with both the simulation and filter configured to utilize a 6.0° AOA/AOE uncertainty. A comparison between the host vehicle position accuracy achieved with a 6.0° AOA/AOE uncertainty versus a 0.7° AOA/AOE uncertainty is presented as Figure 7-43.

For this test case, high quality navigation aids were selected to limit the position error contribution of the navigation aids. Despite this, Figure 7-43 reveals a substantial degradation in all aspects of filter performance when an AOA/AOE uncertainty of 6.0° is used. As indicated in Table 7-7 and Table 7-8, the mean host vehicle EPU was 4,890 meters, the mean RPE mode was 1,233 meters, and the mean VEPU was 1,833 meters. This represents a 567% increase in mean EPU, a 478% increase in the mean RPE mode, and a 498% increase in mean VEPU over the corresponding 0.7° AOA/AOE test. This dismal level of accuracy would be of little benefit to the aviation industry.

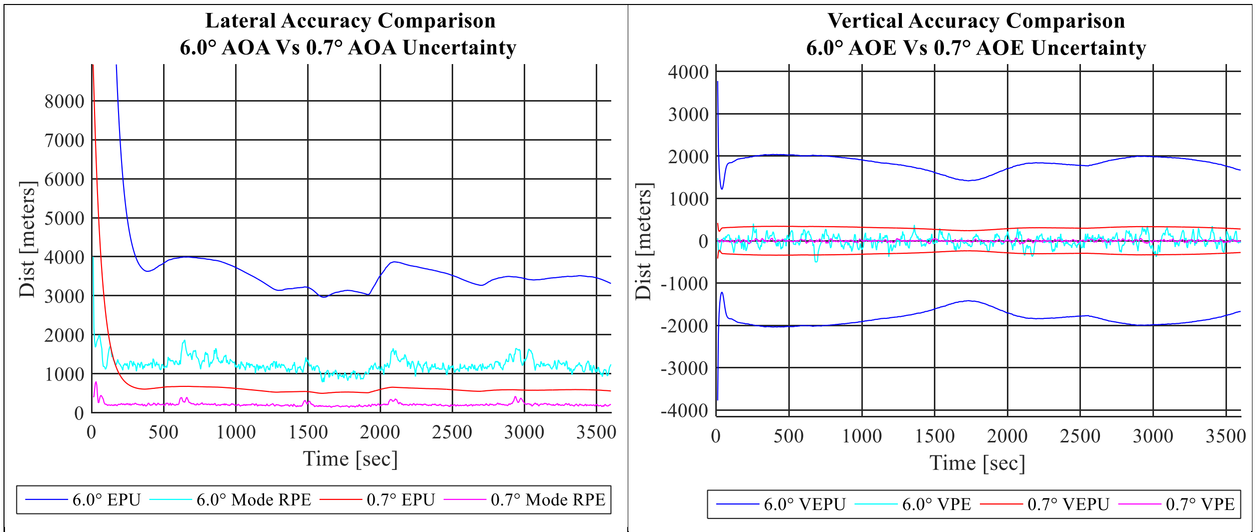

Figure 7-43A: Lateral Performance Comparison. **Figure 7-43B: Vertical Performance Comparison.**

Figure 7-43: Comparison of Filter Performance between 6.0° and 0.7° AOA/AOE Uncertainty.

The performance demonstrated in Figure 7-43 clearly indicates that AOA/AOE measurement uncertainty is the primary contributor to the filter's overall accuracy. This should not come as a surprise because the AOA/AOE measurements constitute the coupling between the navigation aids and the host vehicle position estimate. Any degradation in AOA/AOE measurement performance will naturally affect the overall filter performance. Given this, it may be inferred that improving the AOA/AOE measurement uncertainty below the 0.7° level can be expected to improve the overall accuracy of the algorithm.

7.4 Summary of Results

This section accumulates the summary statistics computed from each test result and condenses them into two tables for easy comparison. Table 7-7 presents the statistics describing the algorithm's lateral performance; while Table 7-8 provides the statistics describing the algorithm's vertical performance.

Table 7-7 and Table 7-8 indicate that the algorithm performed best when an abundance of high quality navigation aids were available. To this end, the best results were achieved with the Random Navigation Aid test, where ten navigation aids were available to the algorithm. For this test, the mean host vehicle EPU was 580 meters and the mean of the host vehicle RPE mode was 162 meters. The most accurate mean host vehicle vertical EPU of 205 meters was also observed during this test. These tables also confirm that navigation aid accuracy is a factor in in the algorithm's overall performance. However, the accuracy of the algorithm is only marginally affected by decreasing NACp until the NACp falls below 6. That is, the filter performance is quite similar for a given number of navigation aids when the NACp is 6 or more. However, the filter performance degrades rapidly as the NACp falls below 6.

Perhaps as important as the actual position computed by the algorithm is the filter's estimate of its position uncertainty. In fact, it could be argued that a position estimate is meaningless without a corresponding uncertainty estimate. Table 7-7 and Table 7-8 demonstrate that the algorithm not only made valid position estimates, it also provided honest uncertainty reports. The average of the reported host vehicle EPU containment of RPE was 97.5%. Likewise, the average of the reported host vehicle VPEU containment of VPE was 96.6%. When compared to the 95% containment objective, it is obvious that the algorithm constructed conservative uncertainty estimates across the full range of test scenarios.

Table 7-7: Summary Statistics for Algorithm's Lateral Position Accuracy.

NACp	GVA	Number Navaids	AOA Sigma [deg]	Average Host EPU [m]	Average Host RPE [m]	Host EPU Containment [%]	Average Navaid Filtered EPU [m]	Average Navaid Filtered RPE [m]	Navaid Filtered EPU Containment [%]
Unk	Unk	<=10	0.7	579.7	162.1	97.4	110.5	39.4	96.4
10	2	8	6.0	4,890	1,233	98.2	80.4	26.2	97.2
10	2	8	0.7	732.8	213.5	97.6	79.6	26.1	96.3
10	2	4	0.7	1,045	292.1	98.1	73.1	24.2	96.9
10	2	2	0.7	2,031	533.2	98.7	69.2	23.5	97.0
10	2	Variable	0.7	1,355	367.2	97.9	88.9	25.5	97.8
6	1	8	0.7	814.3	252.2	97.5	580.3	225.0	96.4
6	1	4	0.7	1,193	344.3	98.1	590.7	221.5	96.7
6	1	2	0.7	2,343	643.7	98.7	611.6	235.6	97.3
6	1	Variable	0.7	1,480	417.5	97.9	592.5	228.6	96.3
2	1	8	0.7	3,822	1,366	97.0	6,476	2,385	95.7
2	1	4	0.7	6,058	2,320	94.8	6,898	2,510	96.5
2	1	2	0.7	13,090	4,199	98.4	7,470	2,999	96.1
2	1	Variable	0.7	6,839	2,454	95.7	6,847	2,617	96.1

Notes: Unk = Unknown. This is due to the random configuration of the navigation aids used for this test.

Variable = Variable number of navigation aids. These tests periodically alter the number of available navigation aids.

Average Host RPE = Average of the Host RPE Modes.

Table 7-8: Summary Statistics for Algorithm's Vertical Position Accuracy.

NACp	GVA	Number Navaids	AOE Sigma [deg]	Average Host VEPU [m]	Average Host VPE [m]	Host VEPU Containment [%]	Average Navaid Filtered VEPU [m]	Average Navaid Filtered VPE [m]	Navaid Filtered VEPU Containment [%]
Unk	Unk	<= 10	0.7	205.0	-5.7	95.9	100.6	-5.7	95.3
10	2	8	6.0	1,833	-4.1	96.4	54.1	-3.9	98.3
10	2	8	0.7	306.5	-4.0	96.7	52.9	-3.4	97.3
10	2	4	0.7	400.1	6.1	96.5	52.9	-4.5	97.9
10	2	2	0.7	579.4	-3.2	96.5	53.4	-2.4	97.1
10	2	Variable	0.7	466.1	-4.8	96.7	53.4	-3.7	98.2
6	1	8	0.7	314.0	-2.9	96.4	149.4	-3.0	94.8
6	1	4	0.7	410.6	-5.5	96.6	151.7	-6.6	96.1
6	1	2	0.7	593.3	-5.2	96.7	159.8	-5.4	96.7
6	1	Variable	0.7	473.4	-1.6	96.3	154.6	-5.1	95.2
2	1	8	0.7	319.8	-5.1	96.3	155.8	-6.5	96.8
2	1	4	0.7	418.8	-4.1	96.4	157.6	-4.4	97.6
2	1	2	0.7	681.5	-6.2	98.1	162.9	-3.5	96.4
2	1	Variable	0.7	496.7	-6.6	96.3	159.9	-5.3	96.9
Notes: Unk = Unknown. This is due to the random configuration of the navigation aids used for this test.									
Variable = Variable number of navigation aids. These tests periodically alter the number of available navigation aids.									

7.5 Key Qualitative Observations

This section provides a summary of some of the key observations noted during analysis of the various test results. Unlike the preceding sections, these observations are mostly qualitative rather than quantitative in nature.

The suite of test cases have demonstrated that the algorithm is fully capable of tracking multiple airborne navigation aids, while simultaneously generating reasonable host vehicle position estimates across a wide variety – yet not exhaustive – set of input conditions. The algorithm has also demonstrated that it responds quickly to changes in the number of available navigation aids without corrupting the host vehicle position estimate. The accuracy of the host vehicle position estimates was shown to be function of the AOA/AOE measurement uncertainty, the number of available navigation aids, the geometry of the navigation aids relative to the host vehicle, and ADS-B reported navigation aid accuracy. The most accurate position estimates were achieved when ten high quality navigation aids were available. In this case, a mean EPU of 580 meters and mean RPE mode of 162 meters were observed. The corresponding VEPU for this test case was 205 meters. Finally, the collection of test scenarios has shown that the filter produces valid – if not somewhat conservative – estimates of its 95th percentile containment levels in the form of EPU and VEPU. Collectively, the average EPU containment was 97.5% and the average VEPU containment was 96.6%.

With respect to tracking the available airborne navigation aids, the algorithm has demonstrated an ability to filter out ADS-B position measurement noise, particularly in the lateral dimension, to generate navigation aid position estimates that are less noisy than the source ADS-B data. It has also been revealed that in certain low quality navigation aid cases, the algorithm is capable of generating navigation aid position estimates that are not only less noisy, but also more accurate than the received ADS-B position reports. Finally, the results indicate that algorithm generates valid filtered navigation aid EPU and VEPU estimates that correctly account for the effects of ADS-B transmission latency. Across the suite of tests, a 96.6% average filtered navigation aid EPU containment level, and an average 96.8% VEPU containment level were observed.

During analysis of the test results, there were a small handful of observations that may warrant additional analysis and potential algorithm updates. These observations are described in the remaining paragraphs of this section.

Analysis of the algorithm's ability to track the vertical position of airborne navigation aids consistently revealed that the filter did little to improve the navigation aid's vertical position estimate. The algorithm performed poorly in rejecting measurement noise and was unable to improve the vertical position estimate. Because the primary goal of the algorithm is to estimate the position of the host vehicle, and by all accounts, the algorithm performed as expected in this regard, the inability of the algorithm to improve the vertical position reports of airborne navigation aids was considered insignificant. The most likely cause of this is the navigation aid vertical dimension process noise setting. But because this setting is embedded in a matrix, and correlated with, the lateral dimensions process noise, adjusting it may also inadvertently affect the filter's ability to track navigation aids in the lateral dimension.

During analysis of the Random Navigation Aids test scenario, it was noted that the mode of the host vehicle RPE and the host vehicle MPU exhibited divergence when the host vehicle experienced a turn. For reference, this was shown in Figure 7-32. Because the RPE mode remained well contained by the EPU, and because precision navigation was not a stated or achievable goal of the algorithm, this behavior was considered acceptable. It may be possible to improve the filter's response by increasing the host vehicle's lateral process noise; however, doing so would likely increase the EPU as well. Because the EPU was generally conservative, additional increases in the EPU were deemed undesirable. It may also be possible to improve this response by implementing an adaptive filter that would automatically adjust the host vehicle process noise when a change in state is detected; however, this was deemed to be beyond the scope of this project.

Throughout the complete series of testing that included 14 Monte-Carlo tests, with each consisting of a 50-run ensemble, there was a single observation that may indicate a potential filter stability issue. On one iteration of the Random Navigation Aids test, the algorithm reported a 15 second discontinuity in the path of a single navigation aid (refer to Section 7.2.3). To date, no explanation of this divergence has been determined. The most likely causes are geometry of the navigation aid relative to the host vehicle, repeated ADS-B data collisions, or other factor that prevented processing of the navigation aid's position reports. The fact that the divergence was relatively short duration, and self-corrected, suggests that this was not an issue with a singularity or numeric precision of the filter, but rather a transient event that led to a temporary loss of the ADS-B data from a single navigation aid.

8.0 CONCLUSION

This section presents a brief overview of the findings discovered during the research described in this paper, then presents ideas for follow on work.

8.1 Summary

The research documented in this paper has demonstrated that a three-dimensional geodetic positioning algorithm based on AOA/AOE measurements from ADS-B capable aircraft is technically feasible given a reliable method from which to determine the AOA/AOE of the received ADS-B signals. The research has shown that the algorithm is fully capable of tracking multiple airborne navigation aids, while simultaneously generating reasonable host vehicle position estimates across a wide variety of anticipated input conditions.

It has been demonstrated that the algorithm is adept at compensating for ADS-B transmission latency and capable of filtering ADS-B position measurement noise to generate navigation aid position estimates that are somewhat less noisy than the source ADS-B data. It has also been shown that due to the cross correlations established in the filter's covariance matrix, in some cases, the algorithm's estimate of an airborne navigation aid's position may be more accurate than the position reported by the navigation aid itself.

The accuracy of the host vehicle position estimate was shown to be a function of the AOA/AOE measurement uncertainty, the number of available navigation aids, the geometry of the navigation aids relative to the host vehicle, and the accuracy of the airborne navigation aids. The most accurate host vehicle position estimates were achieved when ten GNSS quality navigation aids were available to the algorithm. In this case, the mean of the RPE mode was found to be on the order of 165 meters, the mean EPU was on the order of 580 meters, the mean VPE was -6 meters, and the mean VEPU was 205 meters. This implies that the most likely outcome of a similar experiment is a lateral position error on the order of 165 meters and a vertical position error that deviates about the -6 meter level. Finally, analysis of the algorithm's 95th percentile uncertainty estimates established that the algorithm produces somewhat conservative estimates of its uncertainty, making these uncertainty estimates a trustworthy indication of the host vehicle's 95th percentile position accuracy. Taken together, these results indicate that the method is capable of producing a non-precision navigation solution that can guide an aircraft to its destination.

8.2 Future Work

During development of the algorithm, several potential enhancements that were considered beyond the scope of the current research were identified. This section itemizes these potential enhancements for easy reference.

Early in the development of the algorithm it was recognized that utilizing navigation aids with highly accurate velocity estimates (NACv > 1) would occasionally result in significant divergence between the filter's estimate of navigation aid's position and the navigation aid's true position. Allowed to persist, this could ultimately result in an ill-conditioned covariance matrix that could result in a software exception. As a temporary work around, the algorithm has been configured to assume a NACv of 1 for all navigation aids, regardless of received NACv value (refer to Section 4.3.1). Additional diagnostic investigation could be performed to determine the root cause of this filter instability issue, although little improvement in filter performance is expected by improving the navigation aid velocity accuracy input.

As illustrated in the block diagram of Figure 4-4, the algorithm includes logic to reset the filter if the number of available navigation aids falls below two. Although necessary, this elementary logic should be improved. It is clear from the design that a minimum of two navigation aids are required to generate a host vehicle position estimate. However, this is also highly dependent on the geometry of the navigation aids relative to the host vehicle. If the

ADS-B AOA data from the only remaining navigation aids generates an AOI with the host vehicle that is below 20°, then significant errors in the host vehicle position estimate can be expected. This was explained in the context of filter initialization in Section 4.2.7.1. Similar logic should be applied to ensure that at least one pair of navigation aids generates a reasonable AOI with the host vehicle. The filter should be reset in the absence of at least one reasonable pair.

Observation of the algorithm's ability to track the vertical position of airborne navigation aids consistently revealed that the filter did little to improve the navigation aid's vertical position estimate. The algorithm performed poorly in rejecting measurement noise and was unable to improve the vertical position estimate. Because the primary goal of the algorithm is to estimate the position of the host vehicle, and by all accounts, the algorithm performed as expected in this regard, the inability of the algorithm to improve the vertical position reports of airborne navigation aids was considered insignificant. The most likely cause of this are the navigation aid vertical dimension process noise settings. Additional tuning could be performed in an effort to improve the filter's ability to improve navigation aid vertical position estimates, although this is not expected to result in a significant improvement in the host vehicle vertical position estimates.

During analysis of the Random Navigation Aids test scenario, it was noted that the mode of the host vehicle RPE and the host vehicle MPU exhibited divergence when the host vehicle experienced a turn. For reference, this was shown in Figure 7-32. Because the RPE mode remained well contained by the EPU, and because precision navigation was not a stated or achievable goal of the algorithm, this behavior was considered acceptable. It may be possible to improve the filter's response by increasing the host vehicle's lateral process noise; however, doing so would likely increase the EPU as well. Because the EPU was generally conservative, additional increases in the EPU were deemed undesirable. It may also be possible to improve this response by implementing an adaptive filter that would automatically adjust the host vehicle process noise when a change in host vehicle state is detected.

Perhaps the best way to improve the host vehicle position estimate generated by this algorithm would be to implement a sensor fusion design to utilize host vehicle acceleration and or velocity data in the filter. There are several approaches that could be considered for this. Perhaps the most robust solution would be to utilize an inertial navigation system as the basis of the host vehicle position. The algorithm presented in this paper could then be used to estimate the errors in the INS position so that they could be accounted for. A redesign of the filter would be required to achieve this, as the goal of the filter would no longer be to compute the host vehicle position directly, but rather to estimate the errors present in the INS so that the errors could be removed from the INS position.

APPENDIX A – FUNDAMENTAL ROTATION MATRICES

In navigation applications it is often necessary to express a vector given in one coordinate frame in terms of a different coordinate frame. The rotation matrix is the fundamental mechanism used to perform this operation. Given the ubiquitous nature of frame rotation matrices in navigation applications, this Appendix derives the three basic frame rotation matrices used to perform rotations about the X, Y, and Z-axes. The product of these fundamental rotation matrices can then be used to construct higher-order rotation matrices to perform the desired conversion from one coordinate frame to another.

The vector $\boldsymbol{r}$ originates at the origin and extends to the endpoint R as shown in the two-dimensional XY Cartesian coordinate frame of Figure A-1A. The goal is to express the same vector $\boldsymbol{r}$ in terms of the rotated coordinate frame (X'Y'). This is shown in red in Figure A-1B.

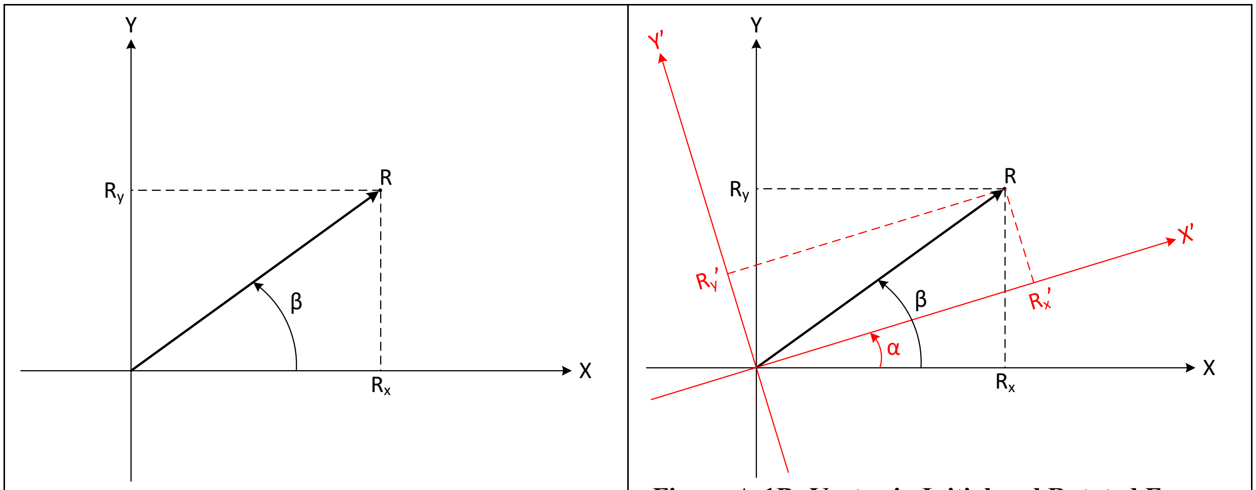

Figure A-1A: Vector in Initial Frame

Figure A-1B: Vector in Initial and Rotated Frame.

Figure A-1: Arbitrary Two-Dimensional Vector Expressed in Original and Rotated Frames.

The coordinate frames are coplanar, with the origin of the rotated frame being coincident with the origin of the initial frame. The two frames are related by the rotation angle α. Basic trigonometry yields the expressions given in (A-1) through (A-4).

$$r_x = |\boldsymbol{r}|\, cos(\beta) \tag{A-1}$$

$$r_y = |\boldsymbol{r}|\, sin(\beta) \tag{A-2}$$

$$r'_x = |\boldsymbol{r}| cos(\beta - \alpha) \tag{A-3}$$

$$r'_y = |\boldsymbol{r}| sin(\beta - \alpha) \tag{A-4}$$

Applying the well-known trigonometric identities for angle differences shown in (A-5) and (A-6) to (A-3) and (A-4) results in (A-7) and (A-8).

$$cos(\beta - \alpha) = cos(\beta)\, cos(\alpha) + sin(\beta) sin(\alpha) \tag{A-5}$$

$$sin(\beta - \alpha) = sin(\beta)\, cos(\alpha) - cos(\beta) sin(\alpha) \tag{A-6}$$

$$r'_x = |\boldsymbol{r}| cos(\beta)\, cos(\alpha) + |\boldsymbol{r}| sin(\beta)\, sin(\alpha) \tag{A-7}$$

$$r'_y = |\boldsymbol{r}|\, sin(\beta)\, cos(\alpha) - |\boldsymbol{r}|\, cos(\beta)\, sin(\alpha) \tag{A-8}$$

Substituting (A-1) and (A-2) into (A-7) and (A-8) results in the expressions for the XY coordinates of $\boldsymbol{r}$, in the rotated frame, as a function of the initial XY coordinates and the rotation angle. This result is shown in (A-9) and (A-10).

$$r'_x = r_x cos(\alpha) + r_y sin(\alpha) \tag{A-9}$$

$$r'_y = -r_x\, sin(\alpha) + r_y cos(\alpha) \tag{A-10}$$

Finally, (A-9) and (A-10) are expressed in the desired matrix form given as (A-11), where $\boldsymbol{C}$ is the rotation matrix. Due to the orthogonal nature of the matrix $\boldsymbol{C}$, the transpose of $\boldsymbol{C}$ results in a rotation in the opposite direction.

$$\boldsymbol{r}' = \boldsymbol{C}(\boldsymbol{r}) = \begin{bmatrix} cos(\alpha) & sin(\alpha) \\ -sin(\alpha) & cos(\alpha) \end{bmatrix} \begin{bmatrix} r_x \\ r_y \end{bmatrix} \tag{A-11}$$

The results for the two-dimensional case can easily be extended to a three-dimensional rotation about the Z-axis as illustrated in Figure A-2. A rotation about the Z-axis implies a rotation from X toward Y in the XY-plane. This is equivalent to the two-dimensional case and these results are given in (A-12) through (A-15), where ***C**z* is the rotation matrix for a positive right handed rotation about the Z-axis.

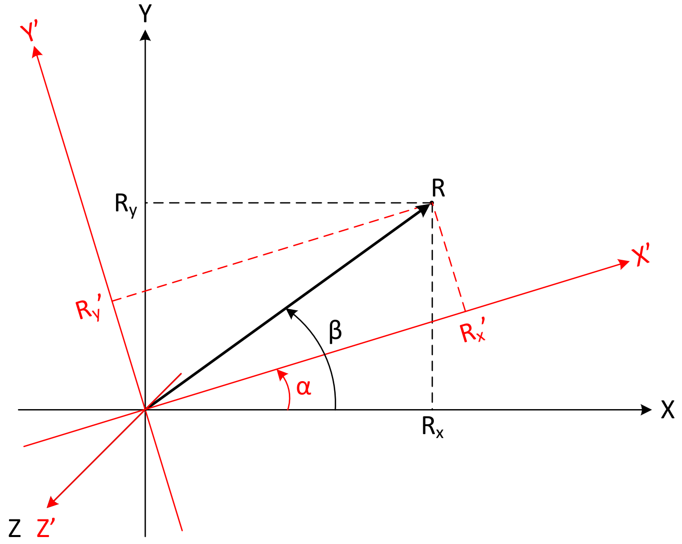

Figure A-2: Three-Dimensional Frame Rotation about the Z-Axis.

$$r_x' = r_x cos(\alpha) + r_y sin(\alpha) \tag{A-12}$$

$$r_y' = -r_x\, sin(\alpha) + r_y cos(\alpha) \tag{A-13}$$

$$r_z' = r_z \tag{A-14}$$

$$\boldsymbol{r}' = \boldsymbol{C_z}(\boldsymbol{r}) = \begin{bmatrix} cos(\alpha) & sin(\alpha) & 0 \\ -sin(\alpha) & cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \tag{A-15}$$

Similarly, a rotation about the X-axis, shown in Figure A-3, implies a frame rotation in the YZ-plane, where the direction of positive right handed rotation is from Y to Z. Following the process for the derivation of a two-dimensional rotation in the XY-plane, with appropriate axis substitutions, it can be shown that the rotation matrix for a rotation about the X-axis is as given in (A-16).

$$\boldsymbol{r}' = \boldsymbol{C_x}(\boldsymbol{r}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha) & sin(\alpha) \\ 0 & -sin(\alpha) & cos(\alpha) \end{bmatrix} \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \tag{A-16}$$

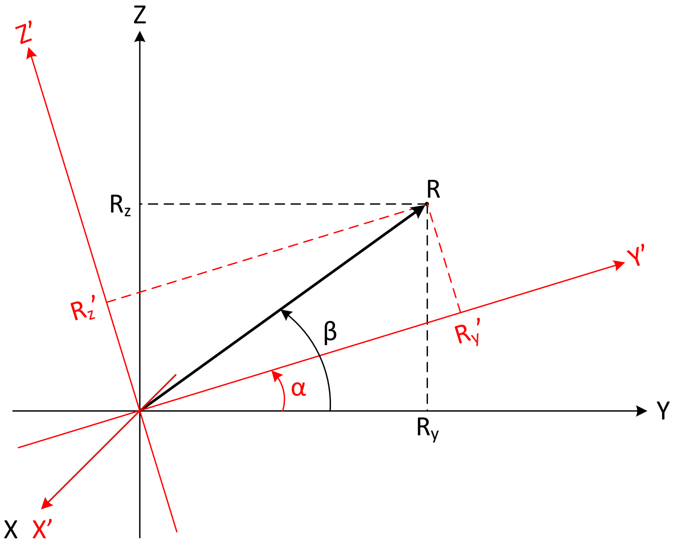

Figure A-3: Three-Dimensional Frame Rotation about the X-Axis.

Finally, a rotation about the Y-axis implies a frame rotation in the ZX-plane, where the direction of positive right handed rotation is from Z to X. The taxonomy of this rotation is given in Figure A-4.

Again following the process for the derivation of the two-dimensional rotation, and again applying appropriate axis substitutions, it can be shown that the rotation matrix for a positive right handed rotation about the Y-axis is as given in (A-26). Note that the position of the negative *sin* term is now at the upper right rather than the lower left. This is due to positive rotation being from the Z-axis to the X-axis. Because this result differs from the others, a summary of the derivation for this rotation is provided in (A-17) through (A-26).

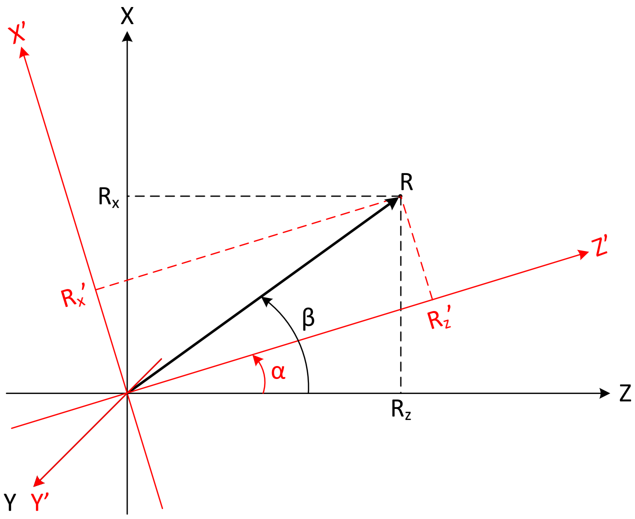

Figure A-4: Three-Dimensional Frame Rotation about the Y-Axis.

$$r_z = |\boldsymbol{r}|\, cos(\beta) \tag{A-17}$$

$$r_x = |\boldsymbol{r}|\, sin(\beta) \tag{A-18}$$

$$r_z' = |\boldsymbol{r}| cos(\beta - \alpha) \tag{A-19}$$

$$r_x' = |\boldsymbol{r}| sin(\beta - \alpha) \tag{A-20}$$

Applying the trigonometric identities for angle differences results in:

$$r_z' = |\boldsymbol{r}| cos(\beta)\, cos(\alpha) + |\boldsymbol{r}| sin(\beta)\, sin(\alpha) \tag{A-21}$$

$$r_x' = |\boldsymbol{r}|\, sin(\beta)\, cos(\alpha) - |\boldsymbol{r}|\, cos(\beta)\, sin(\alpha) \tag{A-22}$$

Back substituting (A-17) and (A-18) into (A-21) and (A-22) results in the coordinates expressed in the rotated frame as given in (A-23) and (A-25). These results are then given in matrix form as (A-26).

$$r_z' = r_z cos(\alpha) + r_x sin(\alpha) \tag{A-23}$$

$$r_x' = -r_z\, sin(\alpha) + r_x cos(\alpha) \tag{A-24}$$

$$r_y' = r_y \tag{A-25}$$

$$\boldsymbol{r}' = \boldsymbol{C}_y(\boldsymbol{r}) = \begin{bmatrix} cos(\alpha) & 0 & -sin(\alpha) \\ 0 & 1 & 0 \\ sin(\alpha) & 0 & cos(\alpha) \end{bmatrix} \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \tag{A-26}$$

APPENDIX B – EARTH CENTERED EARTH FIXED TO NORTH EAST DOWN ROTATION MATRIX

The process of transforming a vector expressed in the ECEF coordinate frame to the equivalent vector in the local geodetic NED frame necessitates the use of a three-dimensional rotation matrix. The current appendix derives this rotation matrix.

The local geodetic NED frame is a locally level navigation frame with an arbitrary origin specified by the origin's geographic latitude and longitude. The relationship between the ECEF frame and an arbitrary NED frame is illustrated in Figure B-1.

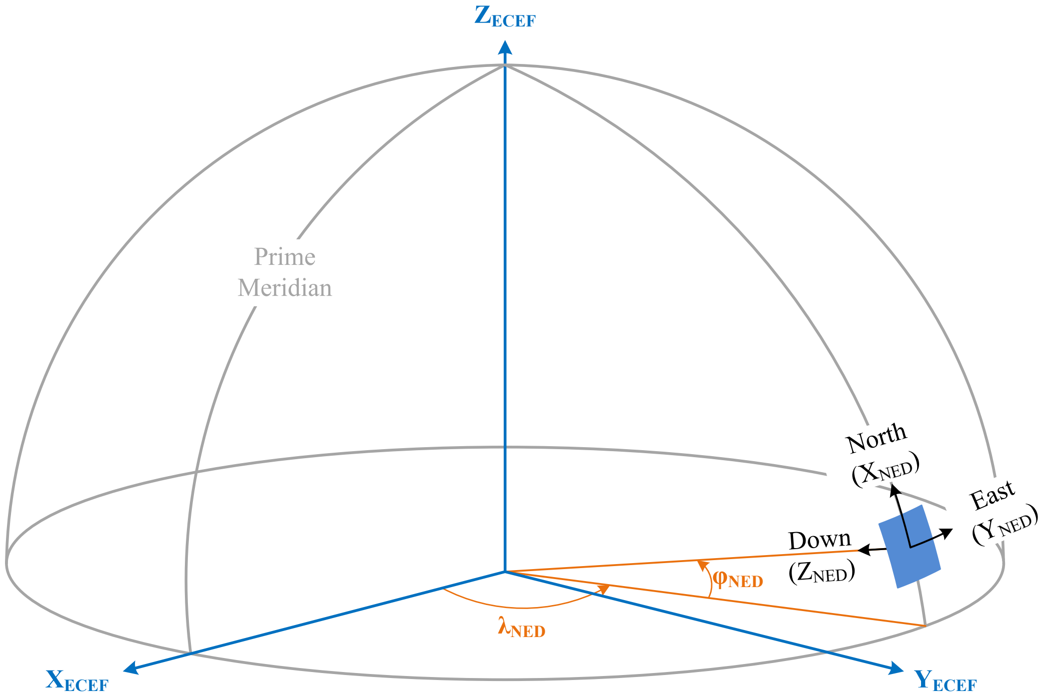

Figure B-1: Nomenclature for the Earth Centered Earth Fixed to North East Down Coordinate Frame Rotation Matrix Derivation.

The rotation matrix to mechanize an ECEF vector in the NED frame is constructed as a series of rotations of the ECEF frame that result in alignment of the rotated ECEF frame with the NED frame. The fundamental rotation matrices required to affect a positive right handed rotation about a specific axis are derived in APPENDIX A. The specific rotations required to achieve the ECEF to NED transformation are:

1. Positive rotation about the ECEF Z-axis (rotating from X towards Y) by the longitude of the NED frame's origin as illustrated in Figure B-2.
2. Negative rotation about the rotated ECEF frame's Y-axis (rotating from X towards Z) by the latitude of the NED frame's origin plus an additional 90° as shown in Figure B-3 and Figure B-4.

The resulting orthogonal rotation matrix is derived in (B-1) through (B-4). Note that the non-commutative nature of matrix multiplication requires that the rotations be expressed in reverse order of application (from right-to-left) in the matrix product.

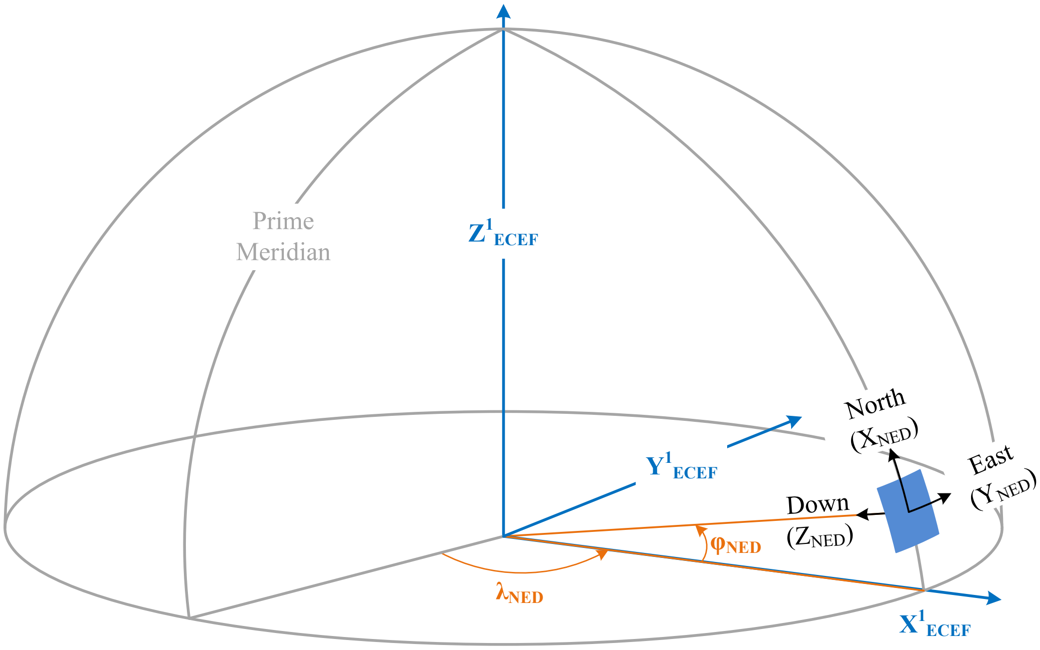

Figure B-2: Earth Centered Earth Fixed Frame following Positive Rotation About the Z-Axis by the Longitude of the North East Down Frame's Origin.

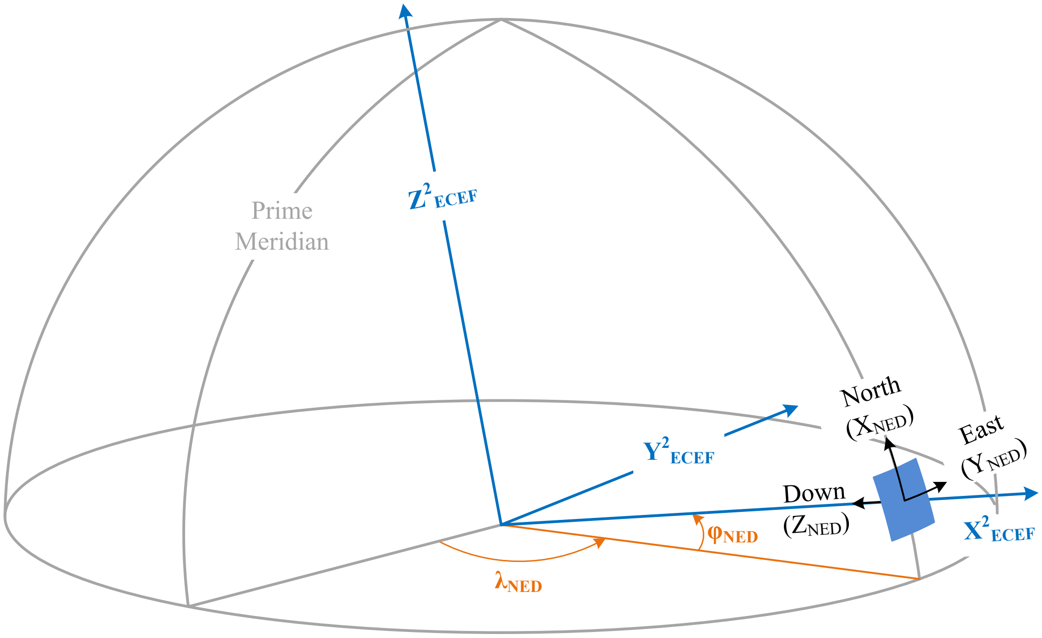

Figure B-3: Earth Centered Earth Fixed Frame following Negative Rotation About the Rotated Y-Axis by the North East Down Frame's Origin Latitude.

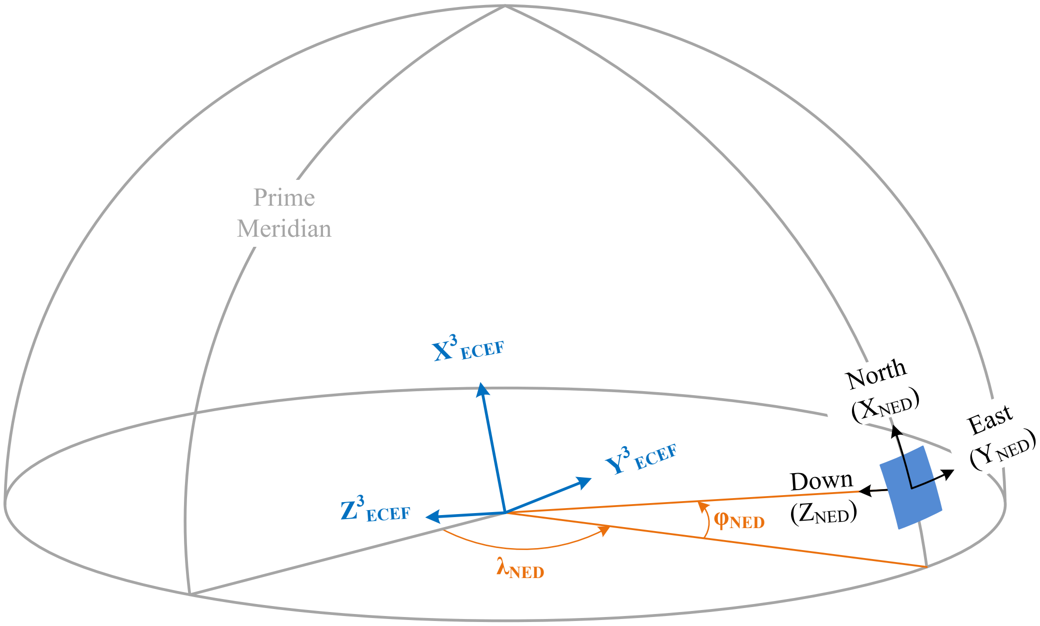

Figure B-4: Earth Centered Earth Fixed Frame following Negative Rotation About the Rotated Y-Axis by 90° Beyond the North East Down Frame's Origin Latitude.

$$C_{ECEF}^{NED} = -R_Y\left(\frac{\pi}{2} + \varphi_{NED}\right) R_Z(\lambda_{NED}) \tag{B-1}$$

Where: C_{ECEF}^{NED} = Coordinate conversion matrix from ECEF to NED.

$-R_Y$ = Negative rotation about the ECEF Y-axis. Refer to APPENDIX A.

R_Z = Positive rotation about the ECEF Z-axis. Refer to APPENDIX A.

φ_{NED} = Latitude of the NED frame’s origin.

λ_{NED} = Longitude of the NED frame’s origin.

$$C_{ECEF}^{NED} = \begin{bmatrix} cos\left(\frac{pi}{2} + \varphi_{NED}\right) & 0 & sin\left(\frac{pi}{2} + \varphi_{NED}\right) \\ 0 & 1 & 0 \\ -sin\left(\frac{pi}{2} + \varphi_{NED}\right) & 0 & cos\left(\frac{pi}{2} + \varphi_{NED}\right) \end{bmatrix} \begin{bmatrix} cos(\lambda_{NED}) & sin(\lambda_{NED}) & 0 \\ -sin(\lambda_{NED}) & cos(\lambda_{NED}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B-2}$$

$$C_{ECEF}^{NED} = \begin{bmatrix} -sin(\varphi_{NED}) & 0 & cos(\varphi_{NED}) \\ 0 & 1 & 0 \\ -cos(\varphi_{NED}) & 0 & -sin(\varphi_{NED}) \end{bmatrix} \begin{bmatrix} cos(\lambda_{NED}) & sin(\lambda_{NED}) & 0 \\ -sin(\lambda_{NED}) & cos(\lambda_{NED}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B-3}$$

$$C_{ECEF}^{NED} = \begin{bmatrix} -sin(\varphi_{NED})\,cos(\lambda_{NED}) & -sin(\varphi_{NED})\,sin(\lambda_{NED}) & cos(\varphi_{NED}) \\ -sin(\lambda_{NED}) & cos(\lambda_{NED}) & 0 \\ -cos(\varphi_{NED})\,cos(\lambda_{NED}) & -cos(\varphi_{NED})\,sin(\lambda_{NED}) & -sin(\varphi_{NED}) \end{bmatrix} \tag{B-4}$$

APPENDIX C – EARTH CENTERED EARTH FIXED TO EAST NORTH UP ROTATION MATRIX

The process of transforming a vector expressed in the ECEF coordinate frame to the equivalent vector in the local geodetic ENU frame requires the use of the three-dimensional rotation matrix derived in this appendix.

The local geodetic ENU frame is a locally level navigation frame with an arbitrary origin specified by the origin's geographic latitude and longitude. The relationship between the ECEF frame and an arbitrary ENU frame is illustrated in Figure C-1.

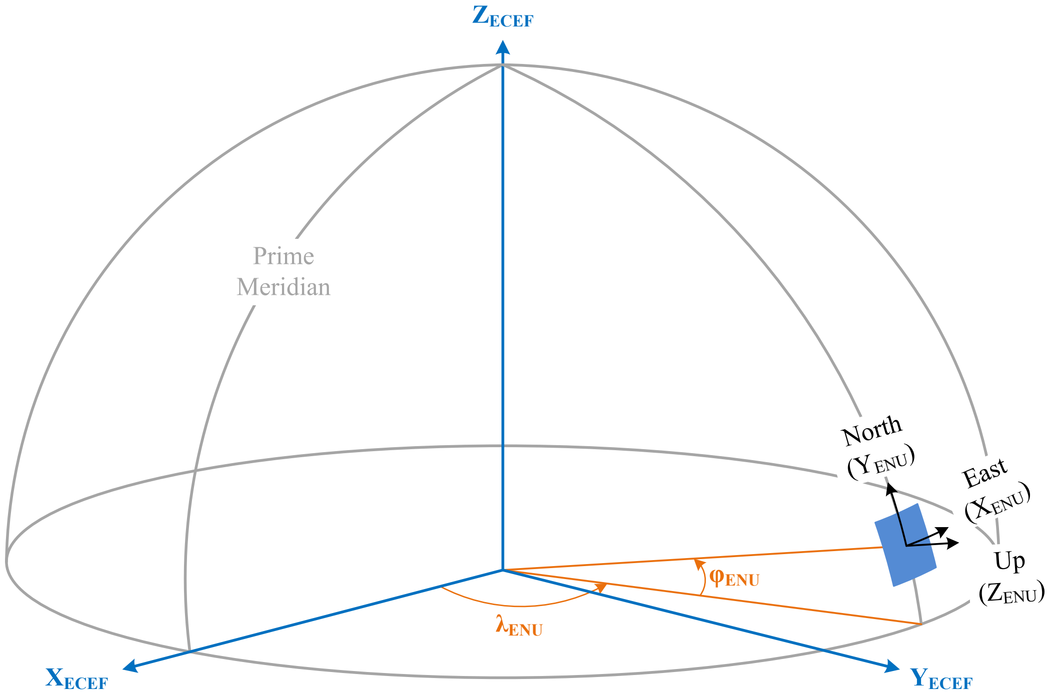

Figure C-1: Taxonomy for the Earth Centered Earth Fixed to East North Up Coordinate Frame Rotation Matrix Derivation.

The rotation matrix to mechanize an ECEF vector in the ENU frame is constructed as a series of rotations of the ECEF frame that result in alignment of the rotated ECEF frame with the ENU frame. The fundamental rotation matrices required to affect a positive right handed rotation about a specific axis are derived in APPENDIX A. The specific rotations required to achieve the ECEF to ENU transformation are:

1. Positive rotation about the ECEF Z-axis (rotating from X towards Y) by the longitude of the ENU frame's origin plus an additional 90° of rotation as illustrated in Figure C-2 and Figure C-3.
2. Positive rotation about the rotated ECEF frame's X-axis (rotating from Y towards Z) by 90° minus the latitude of the ENU frame's origin as shown in Figure C-4.

The resulting orthogonal rotation matrix is derived in (C-1) through (C-4), noting that the non-commutative nature of matrix multiplication requires that the rotations be expressed in reverse order of application (from right-to-left) in the matrix product.

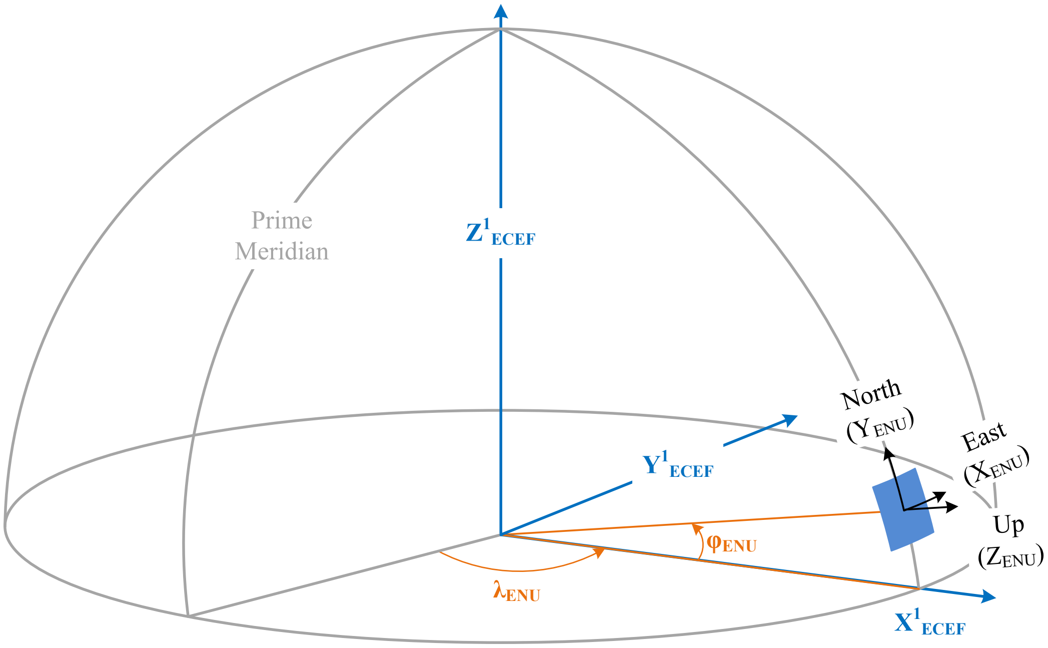

Figure C-2: Earth Centered Earth Fixed Frame following Positive Rotation About the Z-Axis by the Longitude of the East North Up Frame's Origin.

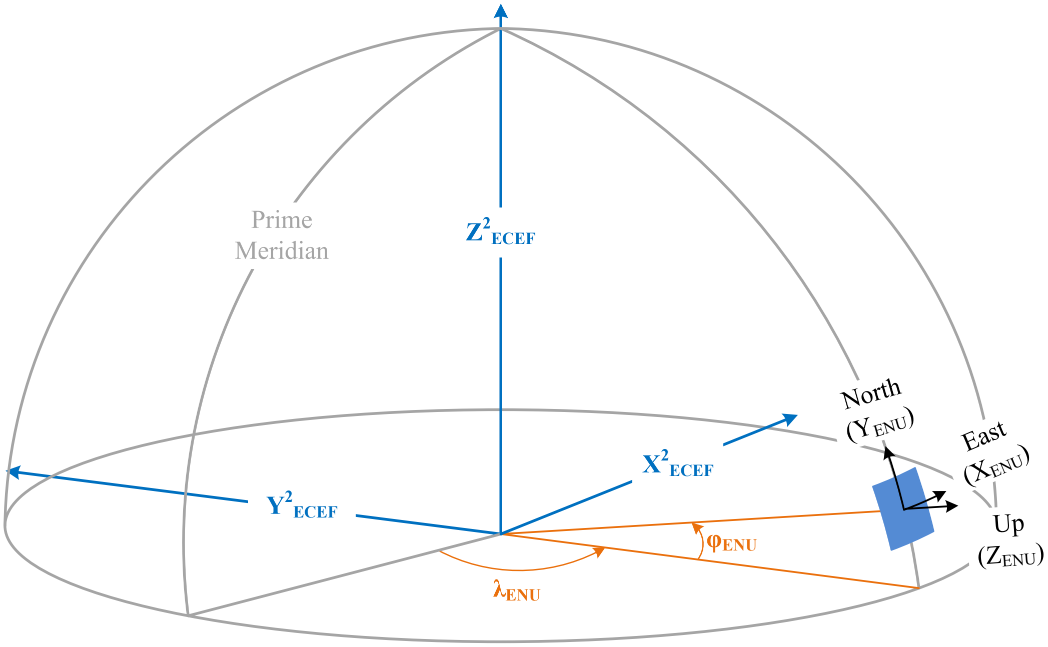

Figure C-3: Earth Centered Earth Fixed Frame following Positive Rotation About the Z-Axis by the Longitude of the East North Up Frame's Origin Plus an Additional 90°.

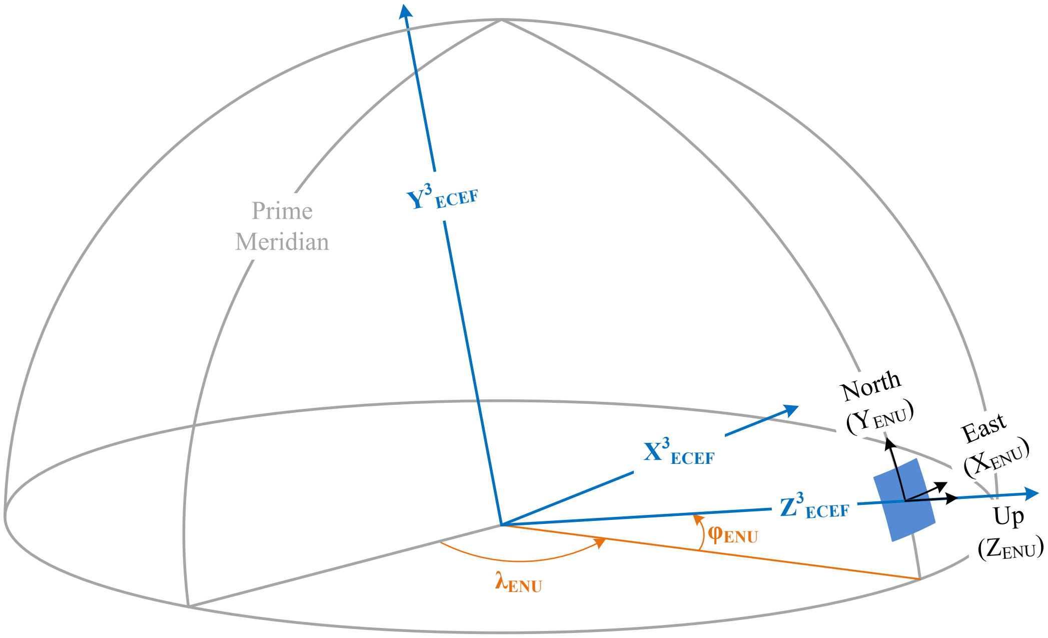

Figure C-4: Earth Centered Earth Fixed Frame following Positive Rotation About the Rotated X-Axis by 90° Minus the Latitude of the East North Up Frame's Origin.

$$C_{ECEF}^{ENU} = R_X\left(\frac{\pi}{2} - \varphi_{ENU}\right) R_Z\left(\frac{\pi}{2} + \lambda_{ENU}\right) \qquad \textbf{(C-1)}$$

Where: C_{ECEF}^{ENU} = Coordinate conversion matrix from ECEF to ENU.

R_X = Positive rotation about the ECEF X-axis. Refer to APPENDIX A.

R_Z = Positive rotation about the ECEF Z-axis. Refer to APPENDIX A.

φ_{ENU} = Latitude of the ENU frame’s origin.

λ_{ENU} = Longitude of the ENU frame’s origin.

$$C_{ECEF}^{ENU} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\left(\frac{pi}{2} - \varphi_{ENU}\right) & sin\left(\frac{pi}{2} - \varphi_{ENU}\right) \\ 0 & -sin\left(\frac{pi}{2} - \varphi_{ENU}\right) & cos\left(\frac{pi}{2} - \varphi_{ENU}\right) \end{bmatrix} \begin{bmatrix} cos\left(\frac{pi}{2} + \lambda_{ENU}\right) & sin\left(\frac{pi}{2} + \lambda_{ENU}\right) & 0 \\ -sin\left(\frac{pi}{2} + \lambda_{ENU}\right) & cos\left(\frac{pi}{2} + \lambda_{ENU}\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \textbf{(C-2)}$$

$$C_{ECEF}^{ENU} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & sin(\varphi_{ENU}) & cos(\varphi_{ENU}) \\ 0 & -cos(\varphi_{ENU}) & sin(\varphi_{ENU}) \end{bmatrix} \begin{bmatrix} -sin(\lambda_{ENU}) & cos(\lambda_{ENU}) & 0 \\ -cos(\lambda_{ENU}) & -sin(\lambda_{ENU}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \textbf{(C-3)}$$

$$C_{ECEF}^{ENU} = \begin{bmatrix} -sin(\lambda_{ENU}) & cos(\lambda_{ENU}) & 0 \\ -cos(\lambda_{ENU})\, sin(\varphi_{ENU}) & -sin(\lambda_{ENU})\, sin(\varphi_{ENU}) & cos(\varphi_{ENU}) \\ cos(\lambda_{ENU})\, cos(\varphi_{ENU}) & sin(\lambda_{ENU})\, cos(\varphi_{ENU}) & sin(\varphi_{ENU}) \end{bmatrix} \qquad \textbf{(C-4)}$$

APPENDIX D – DESCRIPTION OF COVARIANCE MATRIX COORDINATE FRAME CONVERSIONS

When rotating a covariance matrix from one coordinate frame to another, the goal is to utilize a rotation matrix in a similar manner used for position and velocity rotations. Because multiplication by a rotation matrix is a linear combination of both scaling and summation, the resulting equations must satisfy the properties of variances stated in (D-1) and (D-2) [29]. For clarity, (D-2) is expressed explicitly for the sum of two and three random variables in (D-3) and (D-4).

$$var(aX + b) = a^2 var(X) \tag{D-1}$$

Where: a = Scaling factor.

X = Random variable.

$$var\left(\sum_{i=1}^{N} a_i X_i\right) = \sum_{i=1}^{N} a_i^2 var(X_i) + 2 \sum_{1 \le i < j \le N} a_i a_j Cov(X_i, X_j) \tag{D-2}$$

$$var(aX + bY) = a^2 var(X) + b^2 var(Y) + 2abCov(X, Y) \tag{D-3}$$

$$\begin{aligned} &var(aX + bY + cZ) \\ &\quad = a^2 var(X) + b^2 var(Y) + c^2 var(Z) + 2abCov(X, Y) + 2acCov(X, Z) + 2bcCov(Y, Z) \end{aligned} \tag{D-4}$$

The desire is to utilize a function similar to that given in (D-5). Note that (D-5) is for demonstration and development of an intuitive expected outcome. This expression is not mathematically correct because the variance of $\vec{\boldsymbol{X}}$ is not a vector, but rather an *n x n* covariance matrix as given in (D-6) [29]. Expanding the notation, the individual elements of the covariance matrix are defined as shown in (D-7), with a covariance matrix for a 3-dimensional vector of random variables given in (D-8) [29].

$$\boldsymbol{p} = var(\boldsymbol{Cx}) = var\left(\begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}\right) = \begin{bmatrix} var(c_{11}X + c_{12}Y + c_{13}Z) \\ var(c_{21}X + c_{22}Y + c_{23}Z) \\ var(c_{31}X + c_{32}Y + c_{33}Z) \end{bmatrix} \tag{D-5}$$

$$= \begin{bmatrix} c_{11}^2 var(X) + c_{12}^2 var(Y) + c_{13}^2 var(Z) + 2c_{11}c_{12}cov(X,Y) + 2c_{11}c_{13}cov(X,Z) + 2c_{12}c_{13}cov(Y,Z) \\ c_{21}^2 var(X) + c_{22}^2 var(Y) + c_{23}^2 var(Z) + 2c_{21}c_{22}cov(X,Y) + 2c_{21}c_{23}cov(X,Z) + 2c_{22}c_{23}cov(Y,Z) \\ c_{31}^2 var(X) + c_{32}^2 var(Y) + c_{33}^2 var(Z) + 2c_{31}c_{32}cov(X,Y) + 2c_{31}c_{33}cov(X,Z) + 2c_{32}c_{33}cov(Y,Z) \end{bmatrix}$$

$$= \begin{bmatrix} c_{11}^2\sigma_x^2 + c_{12}^2\sigma_y^2 + c_{13}^2\sigma_z^2 + 2c_{11}c_{12}\sigma_{xy}^2 + 2c_{11}c_{13}\sigma_{xz}^2 + 2c_{12}c_{13}\sigma_{yz}^2 \\ c_{21}^2\sigma_x^2 + c_{22}^2\sigma_y^2 + c_{23}^2\sigma_z^2 + 2c_{21}c_{22}\sigma_{xy}^2 + 2c_{21}c_{23}\sigma_{xz}^2 + 2c_{22}c_{23}\sigma_{yz}^2 \\ c_{31}^2\sigma_x^2 + c_{32}^2\sigma_y^2 + c_{33}^2\sigma_z^2 + 2c_{31}c_{32}\sigma_{xy}^2 + 2c_{31}c_{33}\sigma_{xz}^2 + 2c_{32}c_{33}\sigma_{yz}^2 \end{bmatrix}$$

Where: $\boldsymbol{p}$ = The resultant rotated variance vector.
$\boldsymbol{C}$ = Coordinate conversion (rotation) matrix.
$\boldsymbol{x}$ = Vector of random variables X, Y, Z.
σ_i^2 = Variance of $\boldsymbol{x}$ in the *i* dimension.
σ_{ij}^2 = Covariance of $\boldsymbol{x}$ between the *i* and *j* dimensions.

$$var(\boldsymbol{x}) = \boldsymbol{P} = E[(\boldsymbol{x} - E[\boldsymbol{x}])(\boldsymbol{x} - E[\boldsymbol{x}])^T] = E[(\boldsymbol{x} - \boldsymbol{\mu_x})(\boldsymbol{x} - \boldsymbol{\mu_x})^T] \tag{D-6}$$

Where: $\boldsymbol{P}$ = Covariance matrix.

$$var(\boldsymbol{x}) = \boldsymbol{P} = \sum_{i=1}^{N}\sum_{j=1}^{N} cov(x_i, x_j) = \sum_{i=1}^{N}\sum_{j=1}^{N} E\left[(x_i - \mu_i)(x_j - \mu_j)\right] \tag{D-7}$$

$$var(\boldsymbol{x}) = \boldsymbol{P} = \begin{bmatrix} var(x_1) & cov(x_1, x_2) & cov(x_1, x_3) \\ cov(x_1, x_2) & var(x_2) & cov(x_2, x_3) \\ cov(x_1, x_3) & cov(x_2, x_3) & var(x_3) \end{bmatrix} \tag{D-8}$$

The mathematically correct approach to rotate the matrix of covariance values is presented in (D-9) with a proof given as (D-10) [29].

$$\boldsymbol{P'} = Var(\boldsymbol{Cx}) = \boldsymbol{C}Var(\boldsymbol{x})\boldsymbol{C}^T = \boldsymbol{CPC}^T \tag{D-9}$$

Where: $\boldsymbol{P'}$ = Rotated covariance matrix.

$$\begin{aligned} Var(\boldsymbol{Cx}) &= E\{(\boldsymbol{Cx}-\boldsymbol{C\mu})(\boldsymbol{Cx}-\boldsymbol{C\mu})^T\} \\ &= E\{[\boldsymbol{C}(\boldsymbol{x}-\boldsymbol{\mu})][\boldsymbol{C}(\boldsymbol{x}-\boldsymbol{\mu})]^T\} \\ &= E\{[\boldsymbol{C}(\boldsymbol{x}-\boldsymbol{\mu})][(\boldsymbol{x}-\boldsymbol{\mu})]^T\boldsymbol{C}^T\} \\ &= \boldsymbol{C}E\{[(\boldsymbol{x}-\boldsymbol{\mu})][(\boldsymbol{x}-\boldsymbol{\mu})]^T\}\boldsymbol{C}^T \\ &= \boldsymbol{C}var(\boldsymbol{x})\boldsymbol{C}^T \\ &= \boldsymbol{CPC}^T \end{aligned} \quad \textbf{(D-10)}$$

Carrying out the matrix multiplication of (D-9) leads to (D-11). Note that main diagonal elements of (D-11) agree with the intuitive expectations given in (D-5), providing additional confidence in the method. For brevity, the off diagonal elements are not shown, but they express the covariance of the variables in the new reference frame.

$$\begin{aligned} \boldsymbol{P}' &= \boldsymbol{CPC}^T \\ &= \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 & \sigma_{xz}^2 \\ \sigma_{xy}^2 & \sigma_y^2 & \sigma_{yz}^2 \\ \sigma_{xz}^2 & \sigma_{yz}^2 & \sigma_z^2 \end{bmatrix} \begin{bmatrix} c_{11} & c_{21} & c_{31} \\ c_{12} & c_{22} & c_{32} \\ c_{13} & c_{23} & c_{33} \end{bmatrix} = \begin{bmatrix} p'_{11} & p'_{12} & p'_{13} \\ p'_{21} & p'_{22} & p'_{23} \\ p'_{31} & p'_{32} & p'_{33} \end{bmatrix} \end{aligned} \quad \textbf{(D-11)}$$

Where: $p'_{11} = c_{11}^2\sigma_x^2 + c_{12}^2\sigma_y^2 + c_{13}^2\sigma_z^2 + 2c_{11}c_{12}\sigma_{xy}^2 + 2c_{11}c_{13}\sigma_{xz}^2 + 2c_{12}c_{13}\sigma_{yz}^2$

$p'_{22} = c_{21}^2\sigma_x^2 + c_{22}^2\sigma_y^2 + c_{23}^2\sigma_z^2 + 2c_{21}c_{22}\sigma_{xy}^2 + 2c_{21}c_{23}\sigma_{xz}^2 + 2c_{22}c_{23}\sigma_{yz}^2$

$p'_{33} = c_{31}^2\sigma_x^2 + c_{32}^2\sigma_y^2 + c_{33}^2\sigma_z^2 + 2c_{31}c_{32}\sigma_{xy}^2 + 2c_{31}c_{33}\sigma_{xz}^2 + 2c_{32}c_{33}\sigma_{yz}^2$

APPENDIX E – DERIVATION FOR COROLLARY SPHERICAL LAW OF COSINES EQUATION

This appendix provides the direct derivation for equation (2-88) and rationale for (2-89). W.M. Smart provided a similar derivation for determining the adjacent angle B and C of the spherical triangle, but did not derive or present a similar equation for the adjacent angle A. Following the approach presented in Smart's work, the derivation shown in this appendix results in equation (2-88). The same approach can then be used to easily derive (2-89).

E.1 Derivation of Corollary Spherical Law of Cosines Equation

The derivation begins with the spherical law of cosines for side a and side b shown in (E-1) and (E-2) respectively.

$$cos(a) = cos(b)\,cos(c) + sin(b)\,sin(c)\,cos(A) \tag{E-1}$$

$$cos(b) = cos(a)\,cos(c) + sin(a)\,sin(c)\,cos(B) \tag{E-2}$$

Rearranging the terms of (E-1) results in (E-3).

$$sin(b)\,sin(c)\,cos(A) = cos(a) - cos(b)\,cos(c) \tag{E-3}$$

Expanding the $\cos(b)$ term in (E-3) using (E-2) leads to (E-4).

$$sin(b)\,sin(c)\,cos(A) = cos(a) - cos(c)\,[cos(a)\,cos(c) + sin(a)\,sin(c)\,cos(B)] \tag{E-4}$$

Algebraic manipulation of (E-4) gives rise to (E-5).

$$sin(b)\,sin(c)\,cos(A) = cos(a)\,[1 - cos^2(c)] - sin(a)cos(c)sin(c)cos(B) \tag{E-5}$$

Applying the Pythagorean trigonometric identity to (E-5) results in (E-6).

$$sin(b)\,sin(c)\,cos(A) = cos(a)\,sin^2(c) - sin(a)\,cos(c)\,sin(c)\,cos(B) \tag{E-6}$$

Dividing both sides of (E-6) by $\sin(c)$ leads directly to (E-7); which is the desired form of the corollary spherical law of cosines for sides and a direct restatement of (2-88).

$$sin(b)\,cos(A) = cos(a)\,sin(c) - sin(a)cos(c)\,cos(B) \tag{E-7}$$

APPENDIX F – DERIVATION OF THE HAVERSINE SOLUTION FOR GREAT CIRCLE DISTANCE

This appendix presents the derivation for the haversine method of computing great circle distance. The haversine method is considered a numerically stable approach to computing the great circle distance. Formally, the haversine function is defined in (F-1) [51].

$$hav(\theta) = sin^2\left(\frac{\theta}{2}\right) = \frac{1 - cos(\theta)}{2} \tag{F-1}$$

Substituting (F-1) into (2-90) leads to (F-2).

$$1 - 2hav(\theta) = cos(\varphi_p)\, cos(\varphi_q)\left[\, 1 - 2hav(\lambda_q - \lambda_p)\right] + sin(\varphi_p)\, sin(\varphi_q) \tag{F-2}$$

Distributing the cosine terms gives rise to (F-3).

$$\begin{aligned} 1 - 2hav(\theta) = {} & cos(\varphi_p)\, cos(\varphi_q) - 2\, cos(\varphi_p)\, cos(\varphi_q)\, hav(\lambda_q - \lambda_p) \\ & + sin(\varphi_p)\, sin(\varphi_q) \end{aligned} \tag{F-3}$$

Applying the trigonometric identity for angle differences shown in (F-4) to (F-3) leads to (F-5).

$$cos(a - b) = cos(a)\, cos(b) + sin(a)\, sin(b) \tag{F-4}$$

$$1 - 2hav(\theta) = cos(\varphi_p - \varphi_q) - 2\, cos(\varphi_p)\, cos(\varphi_q)\, hav(\lambda_q - \lambda_p) \tag{F-5}$$

Invoking the haversine function once more and simplifying leads to (F-6) and (F-7), where (F-7) is the haversine equation for determining the geocentric angle between two points on a sphere [35] [51].

$$1 - 2hav(\theta) = 1 - 2hav(\varphi_p - \varphi_q) - 2\, cos(\varphi_p)\, cos(\varphi_q)\, hav(\lambda_q - \lambda_p) \tag{F-6}$$

$$hav(\theta) = hav(\varphi_p - \varphi_q) + cos(\varphi_p)\, cos(\varphi_q)\, hav(\lambda_q - \lambda_p) \tag{F-7}$$

Because few calculators or computer programing languages include a haversine or inverse haversine function, a computationally identical form can be determined by applying (F-1) to (F-7) as shown in (F-8) [35]. The geocentric angle θ is then found as given in (F-9).

$$sin^2\left(\frac{\theta}{2}\right) = sin^2\left(\frac{\varphi_p - \varphi_q}{2}\right) + cos(\varphi_p)\, cos(\varphi_q)\, sin^2\left(\frac{\lambda_q - \lambda_p}{2}\right) \tag{F-8}$$

$$\theta = 2arcsin\sqrt{sin^2\left(\frac{\varphi_p - \varphi_q}{2}\right) + cos(\varphi_p)\,cos(\varphi_q)\,sin^2\left(\frac{\lambda_q - \lambda_p}{2}\right)} \tag{F-9}$$

Unlike the spherical law of cosines solutions, the haversine solution can calculate the angular distance between points from nearly 180° down to exactly 0° [51]. However, the haversine solution is numerically ill-conditioned if the points P and Q are nearly antipodal [35]. Antipodal points are not a concern for the current application, but for the sake of completeness, the following solution is provided to remove this constraint.

As in the spherical longitude case, application of the four quadrant arctangent function can be used to eliminate ambiguities. To implement this solution, an expression for both the sin and cos of the geocentric angle must be determined. An expression related to the sin of the geocentric angle is readily available from (F-8) as shown in (F-10) [35].

$$sin\left(\frac{\theta}{2}\right) = \sqrt{sin^2\left(\frac{\varphi_p - \varphi_q}{2}\right) + cos(\varphi_p)\,cos(\varphi_q)\,sin^2\left(\frac{\lambda_q - \lambda_p}{2}\right)} \tag{F-10}$$

An expression related to the cos of the geocentric angle can also be determined from (F-8) by applying the Pythagorean identity. This series of operations is presented as equation (F-11).

$$\begin{aligned} 1 - cos^2\left(\frac{\theta}{2}\right) &= sin^2\left(\frac{\varphi_p - \varphi_q}{2}\right) + cos(\varphi_p)\,cos(\varphi_q)\,sin^2\left(\frac{\lambda_q - \lambda_p}{2}\right) \\ \Rightarrow cos^2\left(\frac{\theta}{2}\right) &= 1 - \left[sin^2\left(\frac{\varphi_p - \varphi_q}{2}\right) + cos(\varphi_p)\,cos(\varphi_q)\,sin^2\left(\frac{\lambda_q - \lambda_p}{2}\right)\right] \\ \Rightarrow cos\left(\frac{\theta}{2}\right) &= \sqrt{1 - \left[sin^2\left(\frac{\varphi_p - \varphi_q}{2}\right) + cos(\varphi_p)\,cos(\varphi_q)\,sin^2\left(\frac{\lambda_q - \lambda_p}{2}\right)\right]} \end{aligned} \tag{F-11}$$

Finally, the geocentric angle can be determined using (F-12).

$$tan\left(\frac{\theta}{2}\right) = \frac{sin\left(\frac{\theta}{2}\right)}{cos\left(\frac{\theta}{2}\right)} = 2\,arctan\left(\frac{sin\left(\frac{\theta}{2}\right)}{cos\left(\frac{\theta}{2}\right)}\right) \tag{F-12}$$

The expressions of (F-10) and (F-11) are often simplified by the substitution given in (F-13). Applying this substitution to (F-12) leads directly to (F-14) [51].

$$a = sin^2\left(\frac{\varphi_p - \varphi_q}{2}\right) + cos\left(\varphi_p\right) cos\left(\varphi_q\right) sin^2\left(\frac{\lambda_q - \lambda_p}{2}\right) \tag{F-13}$$

$$\theta = 2\,arctan\left(\frac{\sqrt{a}}{\sqrt{1-a}}\right) \tag{F-14}$$

APPENDIX G – FIRST ORDER GAUSS-MARKOV PROCESS NOISE

The purpose of this appendix is to provide justification for the first order Gauss-Markov process noise variance value provided by Gelb that was given as (4-29) and restated in (G-1) [48].

$$(\sigma_w)^2 = 2\beta(\sigma_v)^2 \qquad \textbf{(G-1)}$$

Where: $(\sigma_w)^2$ = Gauss-Markov process white noise variance.
β = Correlation time = $1/\tau$
$(\sigma_v)^2$ = Gauss-Markov random variable variance.

Gelb gives the continuous time differential equation for a Gauss-Markov process as shown in (G-2) [48].

$$\dot{v} = -\beta v + w \qquad \textbf{(G-2)}$$

Where: v = Random Gauss-Markov variable
$\dot{v}$ = Derivative of random Gauss-Markov variable.
w = White noise.

The linear variance propagation equation given in (G-3) can be used to propagate covariance forward in time [48]. It's is well known that that the steady state the rate of change is zero, leading to (G-4).

$$\dot{\boldsymbol{P}} = \boldsymbol{FP} + \boldsymbol{PF}^T + \boldsymbol{GQG}^T \qquad \textbf{(G-3)}$$

Where: $\dot{\boldsymbol{P}}$ = Rate of change of covariance.
$\boldsymbol{P}$ = Covariance matrix.
$\boldsymbol{F}$ = State transition matrix.
$\boldsymbol{G}$ = Process noise transition matrix.
$\boldsymbol{Q}$ = Process noise covariance matrix.

$$0 = \boldsymbol{FP} + \boldsymbol{PF}^T + \boldsymbol{GQG}^T \qquad \textbf{(G-4)}$$

Associating the continuous time differential equation terms from (G-2) with the terms of (G-4) by letting ***F* = - *β*** and ***G* = *1*** leads to (G-5), which is equivalent to the original form of (G-1).

$$0 = -\beta(\sigma_v)^2 + (\sigma_v)^2 - \beta^T + (\sigma_w)^2 \qquad \textbf{(G-5)}$$

APPENDIX H – DERIVATION OF AOA AND AOE MEASUREMENT EQUATIONS

Leick et al. provided equations for determining the angle of arrival (α) and angle of elevation (β) for positions expressed in the ECEF frame. There were presented as (4-38) and (4-39) [41]. However, derivations of the equations were not provided. Given the germane nature of these equations to this navigation algorithm, derivations are presented in this appendix.

H.1 Earth Centered Earth Fixed Angle of Arrival Derivation

The derivation in this section determines the north referenced angle from one ECEF position to another ECEF position as provided in Leick et al. and given as (4-38).

Begin by assuming that both positions are expressed in the ENU local geodetic coordinate frame (refer to Section 2.6.1.1.3) rather than the ECEF frame, then the north referenced angle from ENU position A to ENU position B can be determined using planar trigonometry as shown in Figure H-1and Equation (H-1).

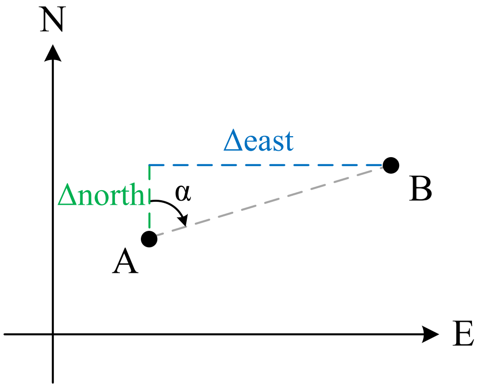

Figure H-1: Angle of Arrival in Arbitrary North/East Frame.

$$tan(\alpha) = \frac{\Delta east}{\Delta north} \tag{H-1}$$

$$\Rightarrow \alpha = tan^{-1}\left(\frac{\Delta east}{\Delta north}\right)$$

Values for $\Delta east$ and $\Delta north$ can easily be determined by rotating the ECEF position delta into the ENU frame using (2-74) (a derivation for this matrix is provided in APPENDIX

C). Expanding the rotation matrix into the corresponding system of equations results in the expressions for $\Delta east$ and $\Delta north$ given in (H-2).

$$\Delta \boldsymbol{r}_{ENU} = \begin{bmatrix} \Delta east \\ \Delta north \\ \Delta up \end{bmatrix} = \boldsymbol{C}_{ECEF}^{ENU} \Delta \boldsymbol{r}_{ECEF} \tag{H-2}$$

$$\Rightarrow \begin{bmatrix} \Delta east \\ \Delta north \\ \Delta up \end{bmatrix} = \begin{bmatrix} -\,sin(\lambda_o) & cos(\lambda_o) & 0 \\ -\,cos(\lambda_o)\,sin(\varphi_o) & -\,sin(\lambda_o)\,sin(\varphi_o) & cos(\varphi_o) \\ cos(\lambda_o)\,cos(\varphi_o) & sin(\lambda_o)\,cos(\varphi_o) & sin(\varphi_o) \end{bmatrix} \begin{bmatrix} \Delta x_{ecef} \\ \Delta y_{ecef} \\ \Delta z_{ecef} \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \Delta east \\ \Delta north \\ \Delta up \end{bmatrix} = \begin{bmatrix} -\,sin(\lambda_o)\Delta x_{ecef} + cos(\lambda_o)\Delta y_{ecef} \\ -\,cos(\lambda_o)\,sin(\varphi_o)\Delta x_{ecef} - sin(\lambda_o)\,sin(\varphi_o)\Delta y_{ecef} + cos(\varphi_o)\Delta z_{ecef} \\ cos(\lambda_o)\,cos(\varphi_o)\Delta x_{ecef} + sin(\lambda_o)\,cos(\varphi_o)\,\Delta y_{ecef} + sin(\varphi_o)\Delta z_{ecef} \end{bmatrix}$$

Substituting $\Delta east$ and $\Delta north$ from (H-2) into (H-1) leads to (H-3), which is the expression for the true north referenced angle between two ECEF positions given in Leick et al in [41].

$$\alpha = tan^{-1}\left[\frac{-\,sin(\lambda_o)\,\Delta x_{ecef} + cos(\lambda_o)\Delta y_{ecef}}{-\,sin(\varphi_o)\,cos(\lambda_o)\,\Delta x_{ecef} - sin(\varphi_o)\,sin(\lambda_o)\,\Delta y_{ecef} + cos(\varphi_o)\Delta z_{ecef}}\right] \tag{H-3}$$

Where: α = True north referenced angle from ECEF position *A* to ECEF position *B*.

φ_o = Latitude of the origin of the local coordinate frame (host vehicle).

λ_o = Longitude of the origin of the local coordinate frame (host vehicle).

$$\Delta \boldsymbol{X}_{ecef} = \begin{bmatrix} \Delta x_{ecef} \\ \Delta y_{ecef} \\ \Delta z_{ecef} \end{bmatrix} = \begin{bmatrix} r_{x_B} - r_{x_A} \\ r_{y_B} - r_{y_A} \\ r_{z_B} - r_{z_A} \end{bmatrix}$$

r_{x,y,z_A} = ECEF position *A*.

r_{x,y,z_B} = ECEF position *B*.

H.2 Earth Centered Earth Fixed Angle of Elevation Derivation

The derivation in this section determines the locally level referenced angle of elevation from one ECEF position to another ECEF position as provided in Leick et al. and given as (4-39).

Again, begin by Assuming that both positions are expressed in the ENU local geodetic coordinate frame (refer to Section 2.6.1.1.3), then the locally level referenced angle of elevation from ENU position A to ENU position B can be determined using planar trigonometry as shown in (H-4).

$$\sin(\beta) = \frac{\Delta up}{|\Delta \boldsymbol{X}_{enu}|} = \frac{\Delta up}{|\Delta \boldsymbol{X}_{ecef}|} \qquad \textbf{(H-4)}$$

Substituting the expression for Δup derived in (H-2) into (H-4) leads directly to the expression provided by Leick et al in [41] and given as (H-5).

$$\beta = sin^{-1}\left[\frac{cos(\varphi_o)\,cos(\lambda_o)\Delta x + cos(\varphi_o)\,sin(\lambda_o)\,\Delta y + sin(\varphi_o)\Delta z}{s}\right] \qquad \textbf{(H-5)}$$

Where: β = Angle of elevation from ECEF position *A* to ECEF position *B*.

φ_o = Latitude of the origin of the local coordinate frame (host vehicle).

λ_o = Longitude of the origin of the local coordinate frame (host vehicle).

s = Magnitude of the vector from position *A* to position *B*.

APPENDIX I – APPLICATION PROGRAM INTERFACE FOR THE ADS-B SIMULATION LIBRARY

The standalone aircraft simulation library simulates ADS-B capable aircraft by providing pseudo ADS-B for user configurable aircraft. The library is designed to be invoked from within a controlling application (like MATLAB). The controlling application is responsible for configuration and control of the simulation library. This arrangement allows the controlling application to invoke the desired number of simulated aircraft, define their flight profiles, establish their navigation accuracy, and control when the aircraft states are recomputed. Configuration of the simulation library is provided through a series of APIs described in this Appendix.

I.1 ADS-B Simulation Top Level Interface

The ADS-B aircraft simulation library must be instantiated before it can be utilized by the controlling application. Instantiation of the library is accomplished by calling the ADSB_Simulation.dll library's class constructor with the parameters shown in Table I-1.

Table I-1: ADS-B Simulation Library Class Constructor.

Function Name:	ADSB_Simulation	
Description:	Instantiates the ADS-B Simulation library.	
Input Parameters	**Data Type**	**Description**
simFreq	Integer	The simulation step frequency in Hz.
acFreq	Integer	The aircraft step frequency in Hz, must be less than or equal to the simulation step frequency.
aoaSigma	Double	The standard deviation of the angle-of-arrival measurement uncertainty.
aoeSigma	Double	The standard deviation of the angle-of-elevation measurement uncertainty.
randomSeed	Boolean	Indicates if the random number generators should be seeded with a random value.

Once instantiated, the simulation and all defined aircraft are controlled by invoking the step function shown in Table I-2. Each call to the step function will advance the simulation one time step and will return a multi-dimensional array that contains the pseudo ADS-B data that would have been received by the host vehicle during that time step. Each row of the array represents a received ADS-B message. The contents of each row vary based on the type of ADS-B message as shown in Table I-3 through Table I-5.

Table I-2: ADS-B Simulation Library Aircraft Step Function.

Function Name:	step	
Description:	Steps the simulation forward in time by one time step.	
Returns	**Data Type**	**Description**
ADSB_Data	Double[][]	Pseudo ADS-B data that would have been received by the host vehicle during the time step.

Table I-3: ADS-B Simulation Library Position Message Array.

Variable	Description
Message Type	Set to 1 to indicate that this is a position message.
ICAO	The ICAO address of the transmitting aircraft.
Receive Time	The time that this message would be received by the host vehicle.
AOA	The angle of arrival for this message.
AOE	The angle of elevation for this message.
Latitude	The received ADS-B position latitude.
Longitude	The received ADS-B position longitude.
Altitude	The received ADS-B position altitude.

Table I-4: ADS-B Simulation Library Velocity Message Array.

Variable	Description
Message Type	Set to 2 to indicate that this is a velocity message.
ICAO	The ICAO address of the transmitting aircraft.
Receive Time	The time that this message would be received by the host vehicle.
AOA	The angle of arrival for this message.
AOE	The angle of elevation for this message.
Velocity East	The received ADS-B east velocity.
Velocity North	The received ADS-B north velocity.
Altitude Rate	The received ADS-B altitude rate.
NACv	The received ADS-B NACv value.

Table I-5: ADS-B Simulation Library Status Message Array.

Variable	Description
Message Type	Set to 3 to indicate that this is a status message.
ICAO	The ICAO address of the transmitting aircraft.
Receive Time	The time that this message would be received by the host vehicle.
AOA	The angle of arrival for this message.
AOE	The angle of elevation for this message.
NACp	The received ADS-B NACp value.
GVA	The received ADS-B GVA value.

I.2 ADS-B Simulation Waypoint Interface

The ADS-B simulation waypoint interface provides the ability to define waypoints that can be used by the simulation for aircraft flight planning and for situational awareness on position output plots generated by the controlling application. Waypoints are defined by calling the waypoint.add function shown in Table I-6.

Table I-6: ADS-B Simulation Library Waypoint Add Procedure.

Function Name:	waypoint.add	
Description:	Adds a waypoint to the ADS-B simulation.	
Input Parameters	**Data Type**	**Description**
ID	String	The unique waypoint identifier.
Latitude	Double	The waypoint latitude in degrees [-90, 90].
Longitude	Double	The waypoint longitude in degrees (-180, 180].

Once added, several functions are available to provide details about the defined waypoints. These function are described in Table I-7 through Table I-9.

Table I-7: ADS-B Simulation Library Waypoint Count Function.

Function Name:	waypoint.count	
Description:	Returns the number of waypoints that exist in the waypoint list.	
Returns	**Data Type**	**Description**
Waypoint Count	Integer	The number of waypoints that have been added.

Table I-8: ADS-B Simulation Library Waypoint Get Function.

Function Name:	waypoint.get	
Description:	Returns the waypoint data for the specified waypoint.	
Input Parameters	**Data Type**	**Description**
Index	Integer	The index of the waypoint to fetch.
Returns	**Data Type**	**Description**
Waypoint Data	WaypointType	The identifier, latitude, and longitude of the waypoint. Latitude and longitude are reported in radians.

Table I-9: ADS-B Simulation Library Waypoint Get Positions Function.

Function Name:	waypoint.getPositions	
Description:	Returns the position information for all defined waypoints.	
Returns	**Data Type**	**Description**
Waypoint Positions	Object[]	A structure containing latitude and longitude. Each structure element is a vector with each entry corresponding to a waypoint. Lat = [lat1, lat2, … lat*N*] Lon = [lon1, lon, … lon*N*]

I.3 ADS-B Simulation Aircraft Interface

The ADS-B simulation aircraft interface provides the ability to configure the simulated aircraft. This interface allows the simulated aircraft to be defined and for current state information to be retrieved for each configured aircraft. The aircraft interface also contains two sub-interfaces. One allows a unique flight plan to be defined for each aircraft and the other allows time based control over the defined aircraft. All of these interfaces are defined in this section.

The top level aircraft interface provides functions/procedures to define the simulated aircraft and to retrieve information pertaining to the current state of each simulated aircraft. Aircraft are defined by invoking one of the two overloads of the aircraft.add function shown in Table I-10 and Table I-11. Current state information can be retrieved using the functions defined in Table I-12 through Table I-16.

Table I-10: ADS-B Simulation Aircraft Add Procedure.

Function Name:	aircraft.add	
Description:	Adds a simulated aircraft to the simulation based on initial latitude/longitude.	
Input Parameters	**Data Type**	**Description**
ICAO	Integer	The unique ICAO address for the aircraft.
Latitude	Double	The initial latitude of the aircraft in degrees [-90, 90].
Longitude	Double	The initial longitude of the aircraft in degrees (-180, 180].
Altitude	Double	The initial altitude of the aircraft in feet.
Speed	Double	The initial speed of the aircraft in knots.
Track	Double	The initial track of the aircraft in degrees.
NACp	Byte	The initial NACp of the aircraft [0, 11].
NACv	Byte	The initial NACv of the aircraft [0, 4].
GVA	Byte	The initial GVA of the aircraft [0, 3].

Table I-11: ADS-B Simulation Aircraft Add Procedure.

Function Name:	aircraft.add	
Description:	Adds a simulated aircraft to the simulation based on initial waypoint position.	
Input Parameters	**Data Type**	**Description**
ICAO	Integer	The unique ICAO address for the aircraft.
Waypoint ID	String	The waypoint identifier to initialize the aircraft to.
Altitude	Double	The initial altitude of the aircraft in feet.
Speed	Double	The initial speed of the aircraft in knots.
Track	Double	The initial track of the aircraft in degrees.
NACp	Byte	The initial NACp of the aircraft [0, 11].
NACv	Byte	The initial NACv of the aircraft [0, 4].
GVA	Byte	The initial GVA of the aircraft [0, 3].

Table I-12: ADS-B Simulation Aircraft Count Function.

Function Name:	aircraft.count	
Description:	Returns the number of aircraft that have been defined.	
Returns	**Data Type**	**Description**
Aircraft Count	Integer	The number of aircraft that have been added.

Table I-13: ADS-B Simulation Aircraft Get Function.

Function Name:	aircraft.get	
Description:	Returns the aircraft state data for the specified aircraft.	
Input Parameters	**Data Type**	**Description**
Index	Integer	The index of the aircraft to fetch.
Returns	**Data Type**	**Description**
Aircraft Data	SimAirplane	A structure of aircraft information including: ICAO, true aircraft state, uncertain aircraft state, reported aircraft position, NACp, NACv, and GVA information.

Table I-14: ADS-B Simulation Aircraft Get Truth State Function.

Function Name:	aircraft.getTruthState	
Description:	Returns the truth state information for all defined aircraft.	
Returns	**Data Type**	**Description**
Aircraft State	Object[]	A structure of: ICAO, latitude, longitude, altitude, speed, and altitude rate. Each structure element is a vector of data. ICAO = [icao1, icao2, … icao*N*] Lat = [lat1, lat2, … lat*N*] etc.

Table I-15: ADS-B Simulation Aircraft Get Uncertain State Function.

Function Name:	aircraft.getUncertainState	
Description:	Returns the uncertain (noisy) state information for all defined aircraft.	
Returns	**Data Type**	**Description**
Aircraft State	Object[]	A structure of: ICAO, latitude, longitude, altitude, speed, and altitude rate. Each structure element is a vector of data. ICAO = [icao1, icao2, … icao*N*] Lat = [lat1, lat2, … lat*N*] etc.

Table I-16: ADS-B Simulation Aircraft Get Reported State Function.

Function Name:	aircraft.getReportedState	
Description:	Returns the state that would be reported in the ADS-B data if the ADS-B were sent at the current time step.	
Returns	**Data Type**	**Description**
Aircraft State	Object[]	A structure of: ICAO, latitude, longitude, altitude, speed, and altitude rate. Each structure element is a vector of data. ICAO = [icao1, icao2, … icao*N*] Lat = [lat1, lat2, … lat*N*] etc.

The aircraft command interface provides time based control over the defined aircraft using the procedures defined in Table I-17 through Table I-22.

Table I-17: ADS-B Simulation Aircraft Command Altitude Function.

Function Name:	aircraft.command.alt	
Description:	Adds an aircraft altitude change command to the aircraft command list.	
Input Parameters	**Data Type**	**Description**
ICAO	Integer	The unique ICAO address for the aircraft.
Time	Double	The time in seconds at which the command is to be applied.
Altitude	Double	The new altitude target for the aircraft. Expressed in feet.

Table I-18: ADS-B Simulation Aircraft Command GVA Function.

Function Name:	aircraft.command.gva	
Description:	Adds an aircraft GVA change command to the aircraft command list.	
Input Parameters	**Data Type**	**Description**
ICAO	Integer	The unique ICAO address for the aircraft.
Time	Double	The time in seconds at which the command is to be applied.
GVA	Byte	The new GVA value [0, 3].

Table I-19: ADS-B Simulation Aircraft Command NACp Function.

Function Name:	aircraft.command.nacP	
Description:	Adds an aircraft NACp change command to the aircraft command list.	
Input Parameters	**Data Type**	**Description**
ICAO	Integer	The unique ICAO address for the aircraft.
Time	Double	The time in seconds at which the command is to be applied.
NACp	Byte	The new NACp value [0, 11].

Table I-20: ADS-B Simulation Aircraft Command NACv Function.

Function Name:	aircraft.command.nacV	
Description:	Adds an aircraft NACv change command to the aircraft command list.	
Input Parameters	**Data Type**	**Description**
ICAO	Integer	The unique ICAO address for the aircraft.
Time	Double	The time in seconds at which the command is to be applied.
NACv	Byte	The new NACv value [0, 4].

Table I-21: ADS-B Simulation Aircraft Command Speed Function.

Function Name:	aircraft.command.spd	
Description:	Adds an aircraft speed change command to the aircraft command list.	
Input Parameters	**Data Type**	**Description**
ICAO	Integer	The unique ICAO address for the aircraft.
Time	Double	The time in seconds at which the command is to be applied.
Speed	Double	The new speed target for the aircraft. Expressed in knots.

Table I-22: ADS-B Simulation Aircraft Command Track Function.

Function Name:	aircraft.command.trk	
Description:	Adds an aircraft track change command to the aircraft command list.	
Input Parameters	**Data Type**	**Description**
ICAO	Integer	The unique ICAO address for the aircraft.
Time	Double	The time in seconds at which the command is to be applied.
Track	Double	The new track target for the aircraft. Expressed in degrees.

The aircraft flight plan interface provides the ability to define a pre-programmed flight profile for each defined aircraft. The flight profile is defined by adding flight plan waypoints for each aircraft to follow. Once the flight plan is defined and engaged, the aircraft will autonomously follow the defined profile. Flight plan waypoints are added by calling one of the four overloads of the function aircraft.flightPlan.addWpt shown in Table I-23 through Table I-26. The flight plan is engaged or disengaged by calling the engage or disengage functions given in Table I-27 and Table I-28 respectively.

Table I-23: ADS-B Simulation Aircraft Flight Plan Add Waypoint Function.

Function Name:	aircraft.flightPlan.addWpt	
Description:	Adds a flight plan waypoint to the aircraft's flight plan. Speed and altitude targets are updated after the waypoint is sequenced.	
Input Parameters	**Data Type**	**Description**
ICAO	Integer	The unique ICAO address for the aircraft.
Waypoint ID	String	The identifier of a predefined simulation waypoint (Section I.2).
Next Altitude	Double	The altitude target (feet) to apply after sequencing the waypoint.
Next Speed	Double	The speed target (knots) to apply after sequencing the waypoint.

Table I-24: ADS-B Simulation Aircraft Flight Plan Add Waypoint Function.

Function Name:	aircraft.flightPlan.addWpt	
Description:	Adds a flight plan waypoint to the aircraft's flight plan. Speed and altitude targets are updated after the waypoint is sequenced.	
Input Parameters	**Data Type**	**Description**
ICAO	Integer	The unique ICAO address for the aircraft.
Latitude	Double	The latitude (degrees) of the waypoint [-90, 90].
Longitude	Double	The longitude (degrees) of the waypoint (-180, 180].
Next Altitude	Double	The altitude target (feet) to apply after sequencing the waypoint.
Next Speed	Double	The speed target (knots) to apply after sequencing the waypoint.

Table I-25: ADS-B Simulation Aircraft Flight Plan Add Waypoint Function.

Function Name:	aircraft.flightPlan.addWpt	
Description:	Adds a flight plan waypoint to the aircraft's flight plan. Speed and altitude targets are not changed when the waypoint is sequenced.	
Input Parameters	**Data Type**	**Description**
ICAO	Integer	The unique ICAO address for the aircraft.
Waypoint ID	String	The identifier of a predefined simulation waypoint (Section I.2).

Table I-26: ADS-B Simulation Aircraft Flight Plan Add Waypoint Function.

Function Name:	aircraft.flightPlan.addWpt	
Description:	Adds a flight plan waypoint to the aircraft's flight plan. Speed and altitude targets are not changed when the waypoint is sequenced.	
Input Parameters	**Data Type**	**Description**
ICAO	Integer	The unique ICAO address for the aircraft.
Latitude	Double	The latitude (degrees) of the waypoint [-90, 90].
Longitude	Double	The longitude (degrees) of the waypoint (-180, 180].

Table I-27: ADS-B Simulation Aircraft Flight Plan Engage Function.

Function Name:	aircraft.flightPlan.engage	
Description:	Engages autonomous control to the aircraft's defined flight plan.	
Input Parameters	**Data Type**	**Description**
ICAO	Integer	The unique ICAO address for the aircraft.
Time	Double	The time (seconds) at which to engage the flight plan.

Table I-28: ADS-B Simulation Aircraft Flight Plan Disengage Function.

Function Name:	aircraft.flightPlan.disengage	
Description:	Disengages autonomous control to the aircraft's defined flight plan.	
Input Parameters	**Data Type**	**Description**
ICAO	Integer	The unique ICAO address for the aircraft.
Time	Double	The time (seconds) at which to disengage the flight plan.

APPENDIX J – PARAMETRIC TEST CASE RESULTS

This appendix provides the graphical results of the Parametric Test Cases described in Table 6-3 of Section 6.3. Given the large number of parametric test cases, only a summary of the results was provided in Sections 7.3 and 7.4. This appendix provides detailed plots of the results from which the summary statistics were computed. Due to the fact that numerous examples were presented to demonstrate the correct and expected behavior of the simulation, plots of the reported ADS-B positions are not provided in this appendix. Instead, this appendix focuses on the plots that demonstrate the behavior of the filter.

J.1 Parametric Test Case 1: 8 Navigation Aids, NACp 10, GVA 2, 6° AOA Uncertainty

The summary statistics for Parametric Test Case 1 are provided in Table J-1, with supporting data plots provided in Figure J-1 through Figure J-4.

Table J-1: Summary Statistics for Parametric Test Case 1.
8 Navigation Aids, NACp 10, GVA 2, 6° AOA/AOE Uncertainty.

Summary Statistic	Units	Value	Summary Statistic	Units	Value
Average Host EPU	[m]	4,890	Average Navaid EPU	[m]	110.5
Average Host RPE Mode	[m]	1,233	Average Navaid RPE Mode	[m]	39.4
Host EPU Containment	[%]	98.2	Navaid EPU Containment	[%]	96.4
Average Host VEPU	[m]	1,833	Average Navaid VEPU	[m]	54.1
Average Host VPE	[m]	-4.1	Average Navaid VPE	[m]	-3.9
Host VEPU Containment	[%]	96.4	Navaid VEPU Containment	[%]	98.3

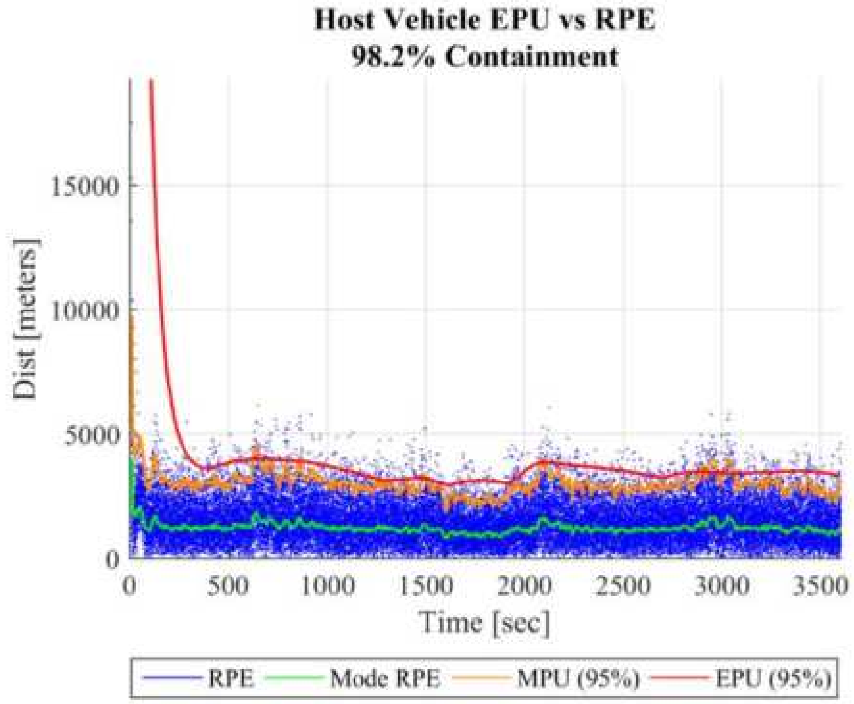

Figure J-1: Host Vehicle Lateral Position Accuracy for the 8 Navigation Aid, NACp 10, GVA 2, 6° AOA Uncertainty Test Case.

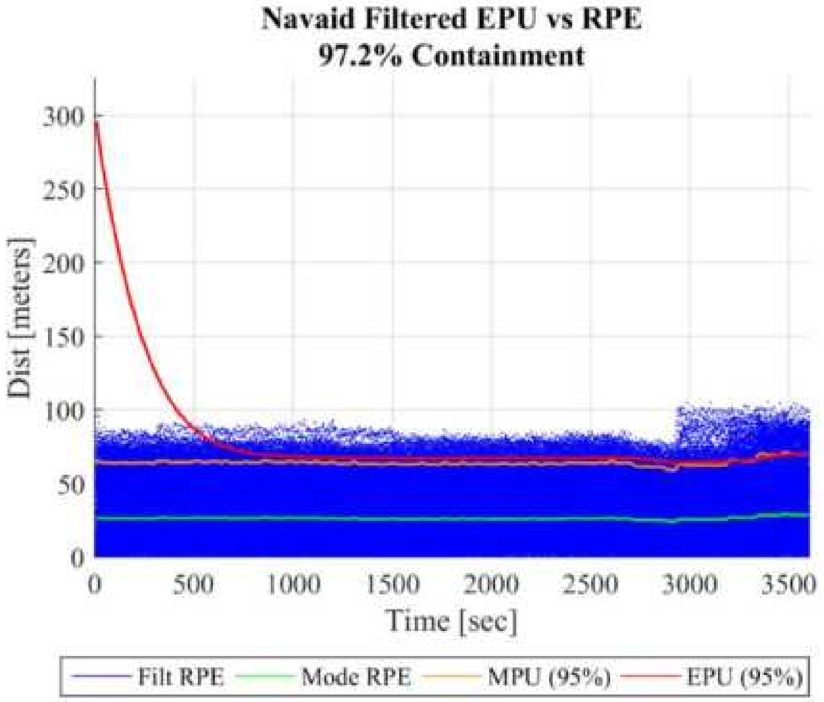

Figure J-2: Navigation Aid Filtered Lateral Position Accuracy for the 8 Navigation Aid, NACp 10, GVA 2, 6° AOA Uncertainty Test Case.

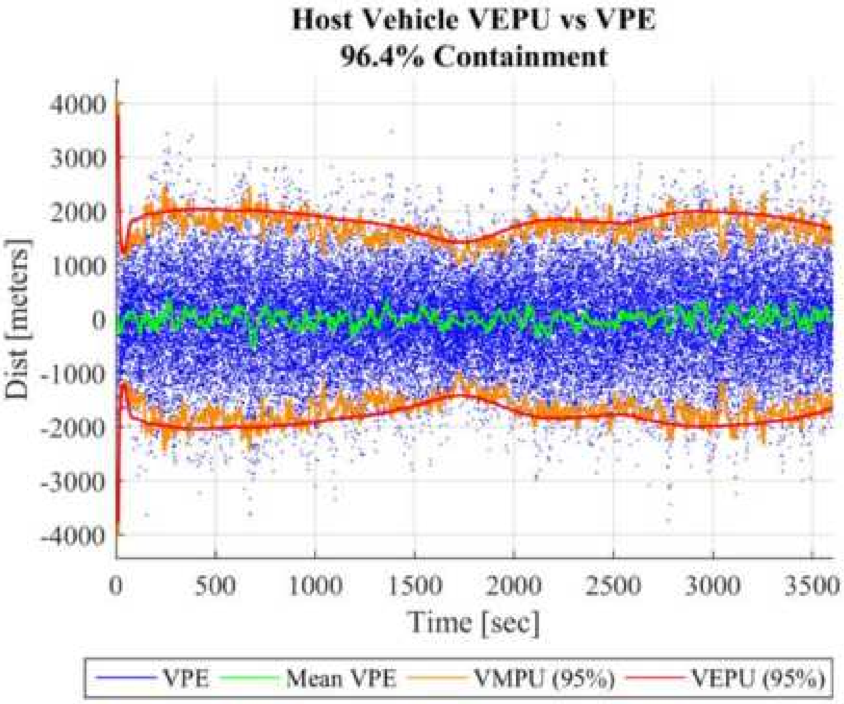

Figure J-3: Host Vehicle Vertical Position Accuracy for the 8 Navigation Aid, NACp 10, GVA 2, 6° AOE Uncertainty Test Case.

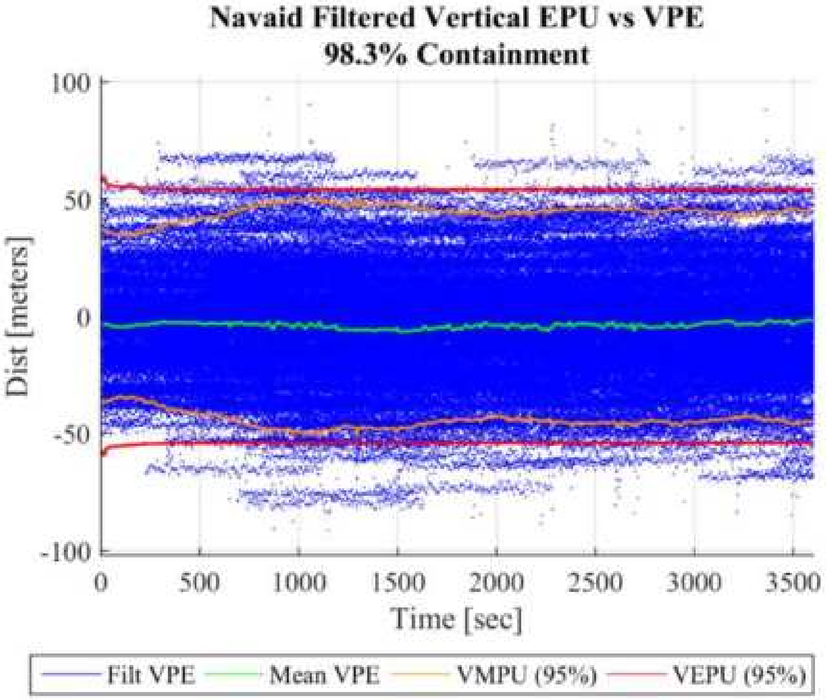

Figure J-4: Navigation Aid Filtered Vertical Position Accuracy for the 8 Navigation Aid, NACp 10, GVA 2, 6° AOE Uncertainty Test Case.

J.2 Parametric Test Case 2: 8 Navigation Aids, NACp 10, GVA 2, 0.7° AOA Uncertainty

The summary statistics for Parametric Test Case 2 are provided in Table J-2, with supporting data plots provided in Figure J-5 through Figure J-8.

Table J-2: Summary Statistics for Parametric Test Case 2.
8 Navigation Aids, NACp 10, GVA 2, 0.7° AOA/AOE Uncertainty.

Summary Statistic	Units	Value	Summary Statistic	Units	Value
Average Host EPU	[m]	732.8	Average Navaid EPU	[m]	79.6
Average Host RPE Mode	[m]	213.5	Average Navaid RPE Mode	[m]	26.1
Host EPU Containment	[%]	97.6	Navaid EPU Containment	[%]	96.3
Average Host VEPU	[m]	306.5	Average Navaid VEPU	[m]	52.9
Average Host VPE	[m]	-4.0	Average Navaid VPE	[m]	-3.4
Host VEPU Containment	[%]	96.7	Navaid VEPU Containment	[%]	97.3

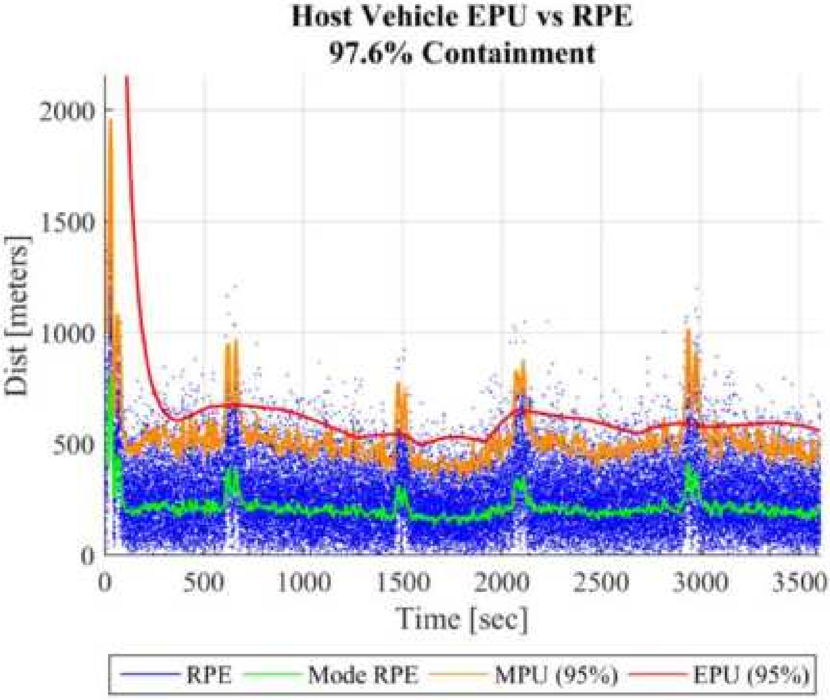

Figure J-5: Host Vehicle Lateral Position Accuracy for the 8 Navigation Aid, NACp 10, GVA 2, 0.7° AOA Uncertainty Test Case.

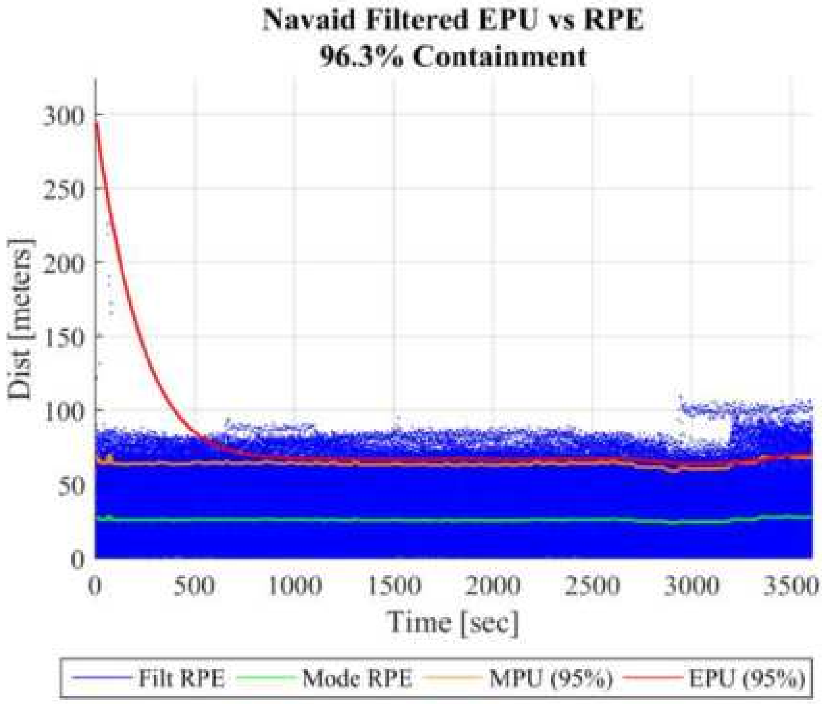

Figure J-6: Navigation Aid Filtered Lateral Position Accuracy for the 8 Navigation Aid, NACp 10, GVA 2, 0.7° AOA Uncertainty Test Case.

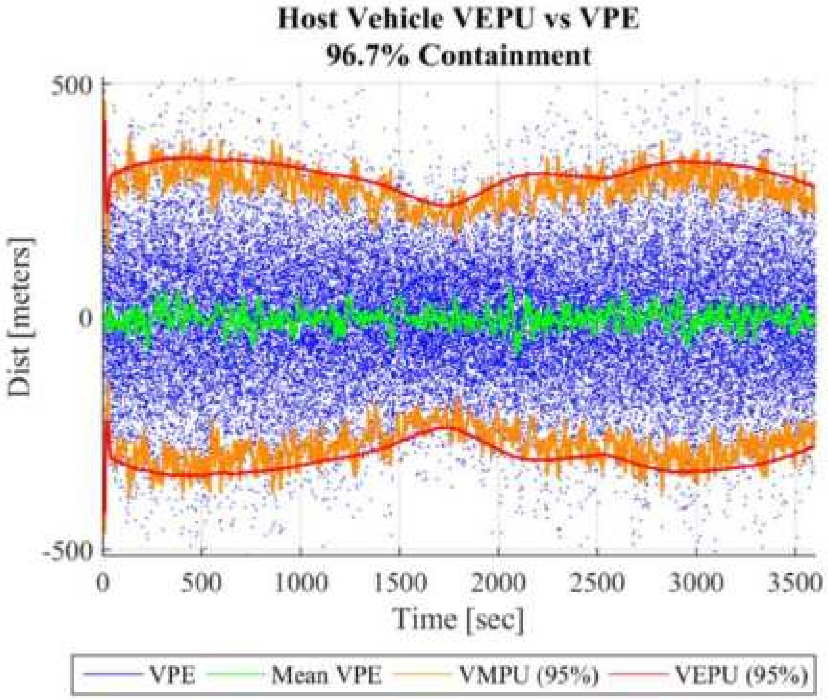

Figure J-7: Host Vehicle Vertical Position Accuracy for the 8 Navigation Aid, NACp 10, GVA 2, 0.7° AOE Uncertainty Test Case.

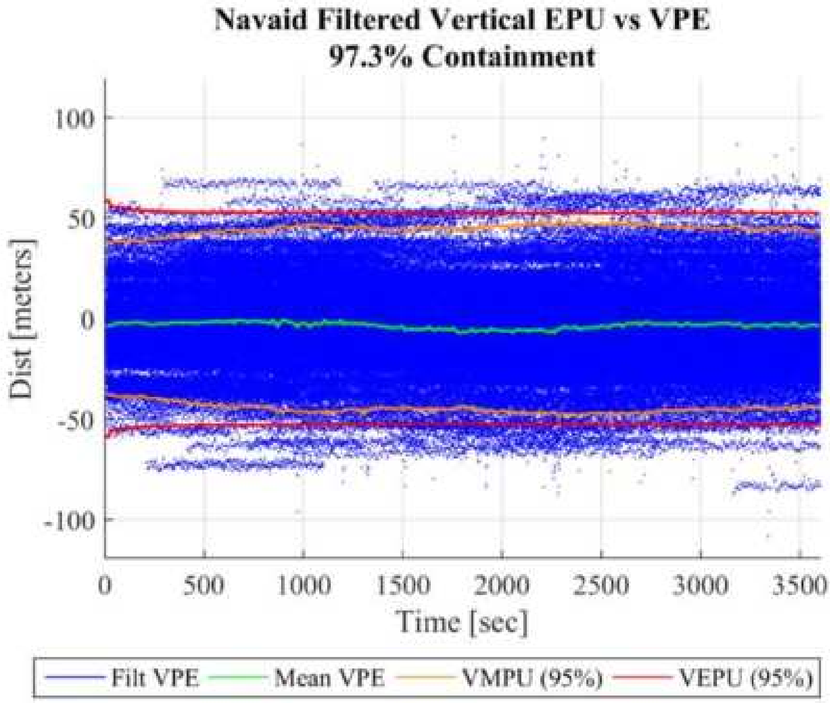

Figure J-8: Navigation Aid Filtered Vertical Position Accuracy for the 8 Navigation Aid, NACp 10, GVA 2, 0.7° AOE Uncertainty Test Case.

J.3 Parametric Test Case 3: 8 Navigation Aids, NACp 6, GVA 1, 0.7° AOA Uncertainty

The summary statistics for Parametric Test Case 3 are provided in Table J-3, with supporting data plots provided in Figure J-9 through Figure J-12.

Table J-3: Summary Statistics for Parametric Test Case 3.
8 Navigation Aids, NACp 6, GVA 1, 0.7° AOA/AOE Uncertainty.

Summary Statistic	Units	Value	Summary Statistic	Units	Value
Average Host EPU	[m]	814.3	Average Navaid EPU	[m]	580.3
Average Host RPE Mode	[m]	252.2	Average Navaid RPE Mode	[m]	225.0
Host EPU Containment	[%]	97.5	Navaid EPU Containment	[%]	96.4
Average Host VEPU	[m]	314.0	Average Navaid VEPU	[m]	149.4
Average Host VPE	[m]	-2.9	Average Navaid VPE	[m]	-3.0
Host VEPU Containment	[%]	96.4	Navaid VEPU Containment	[%]	94.8

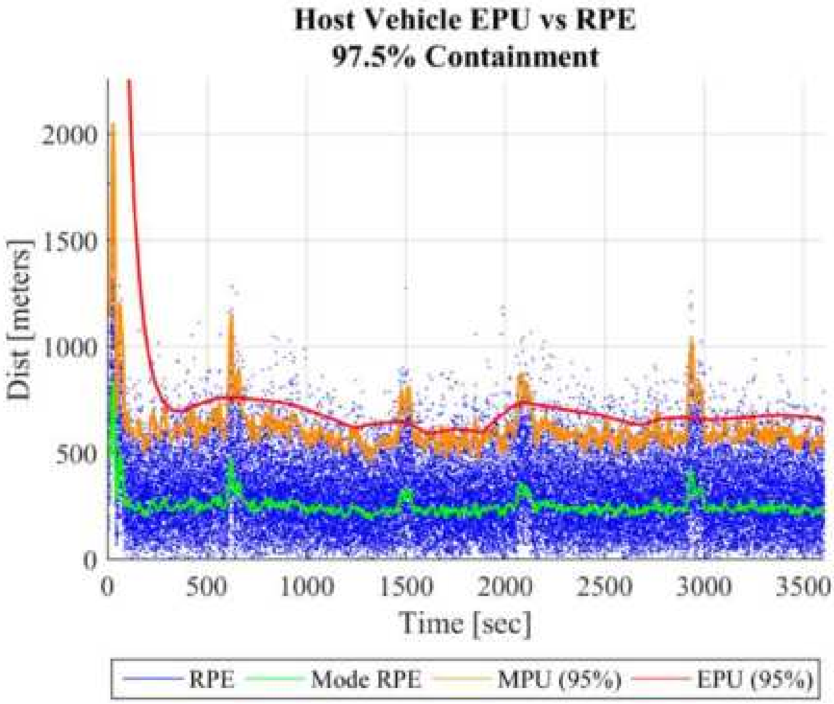

Figure J-9: Host Vehicle Lateral Position Accuracy for the 8 Navigation Aid, NACp 6, GVA 1, 0.7° AOA Uncertainty Test Case.

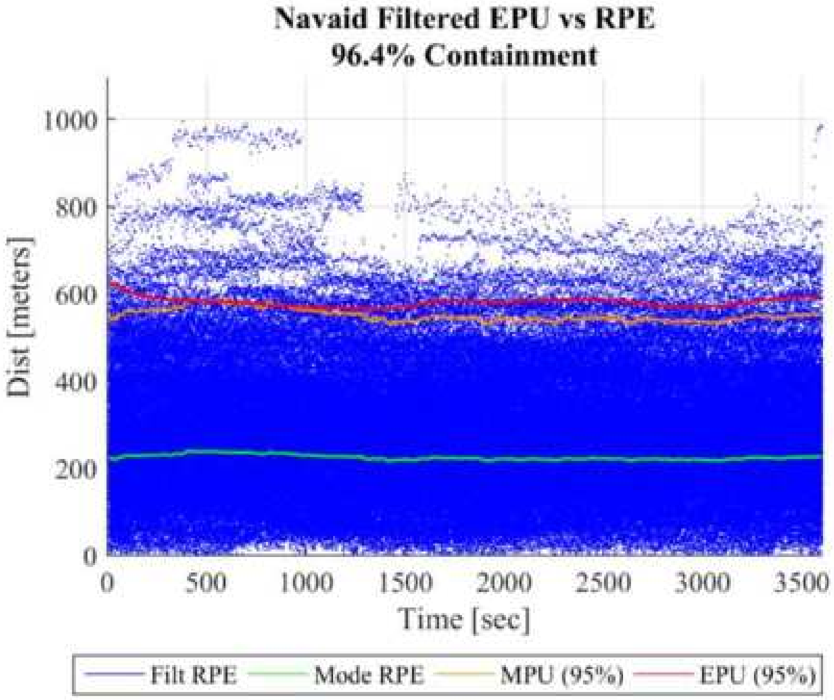

Figure J-10: Navigation Aid Filtered Lateral Position Accuracy for the 8 Navigation Aid, NACp 6, GVA 1, 0.7° AOA Uncertainty Test Case.

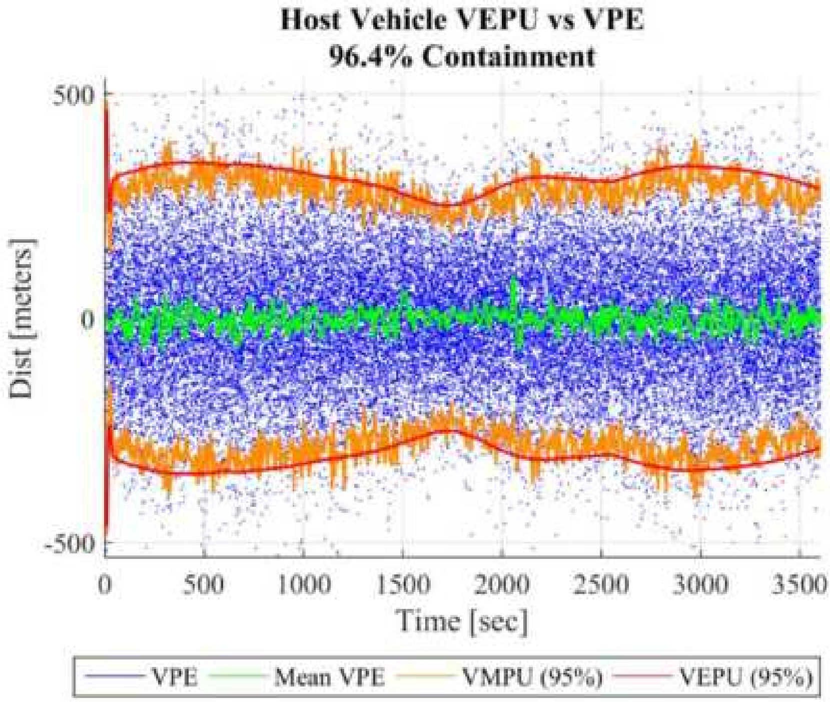

Figure J-11: Host Vehicle Vertical Position Accuracy for the 8 Navigation Aid, NACp 6, GVA 1, 0.7° AOE Uncertainty Test Case.

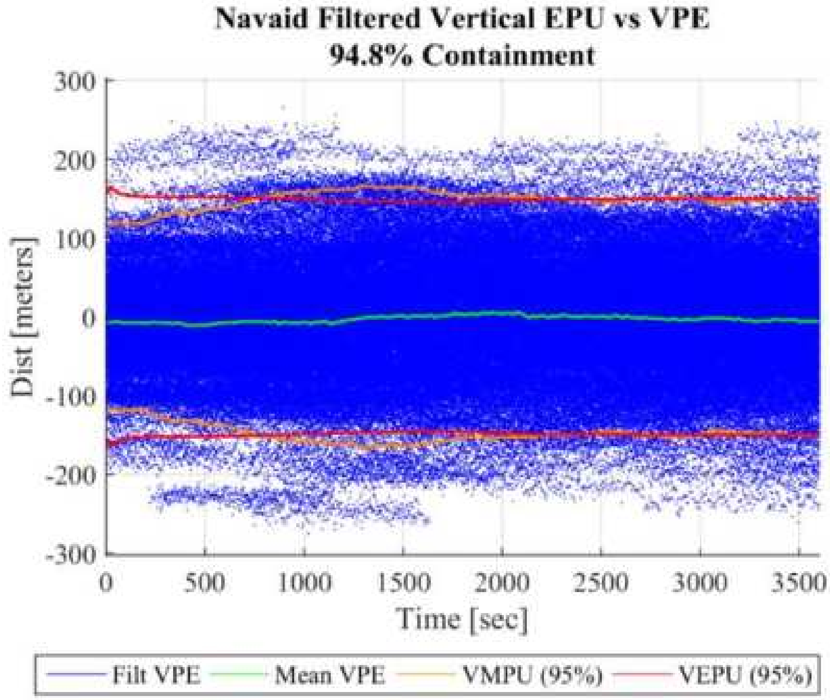

Figure J-12: Navigation Aid Filtered Vertical Position Accuracy for the 8 Navigation Aid, NACp 6, GVA 1, 0.7° AOE Uncertainty Test Case.

J.4 Parametric Test Case 4: 8 Navigation Aids, NACp 2, GVA 1, 0.7° AOA Uncertainty

The summary statistics for Parametric Test Case 4 are provided in Table J-4, with supporting data plots provided in Figure J-13 through Figure J-16.

Table J-4: Summary Statistics for Parametric Test Case 4.
8 Navigation Aids, NACp 2, GVA 1, 0.7° AOA/AOE Uncertainty.

Summary Statistic	Units	Value	Summary Statistic	Units	Value
Average Host EPU	[m]	3,822	Average Navaid EPU	[m]	6,476
Average Host RPE Mode	[m]	1,366	Average Navaid RPE Mode	[m]	2,385
Host EPU Containment	[%]	97.0	Navaid EPU Containment	[%]	95.7
Average Host VEPU	[m]	319.8	Average Navaid VEPU	[m]	155.8
Average Host VPE	[m]	-5.1	Average Navaid VPE	[m]	-6.5
Host VEPU Containment	[%]	96.3	Navaid VEPU Containment	[%]	96.8

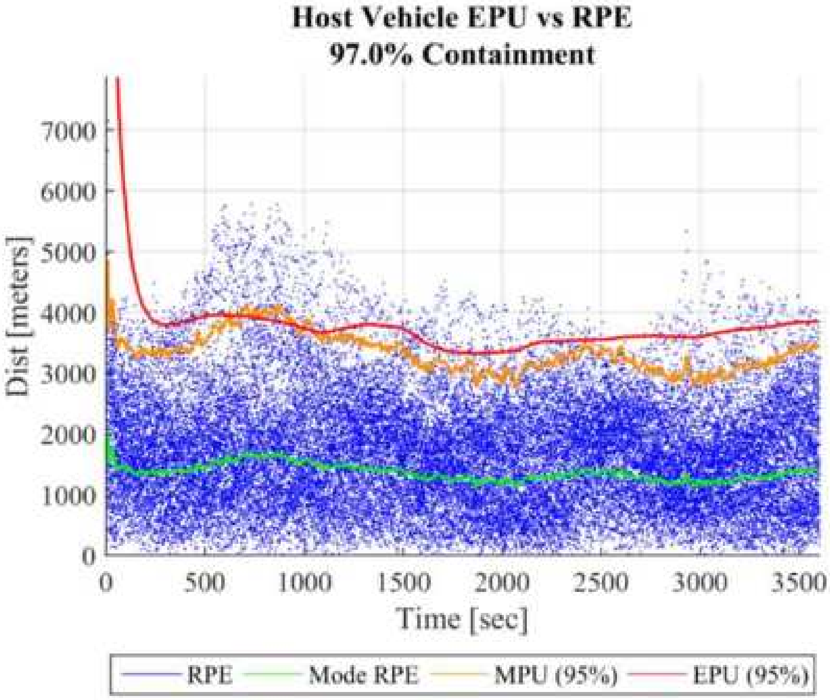

Figure J-13: Host Vehicle Lateral Position Accuracy for the 8 Navigation Aid, NACp 2, GVA 1, 0.7° AOA Uncertainty Test Case.

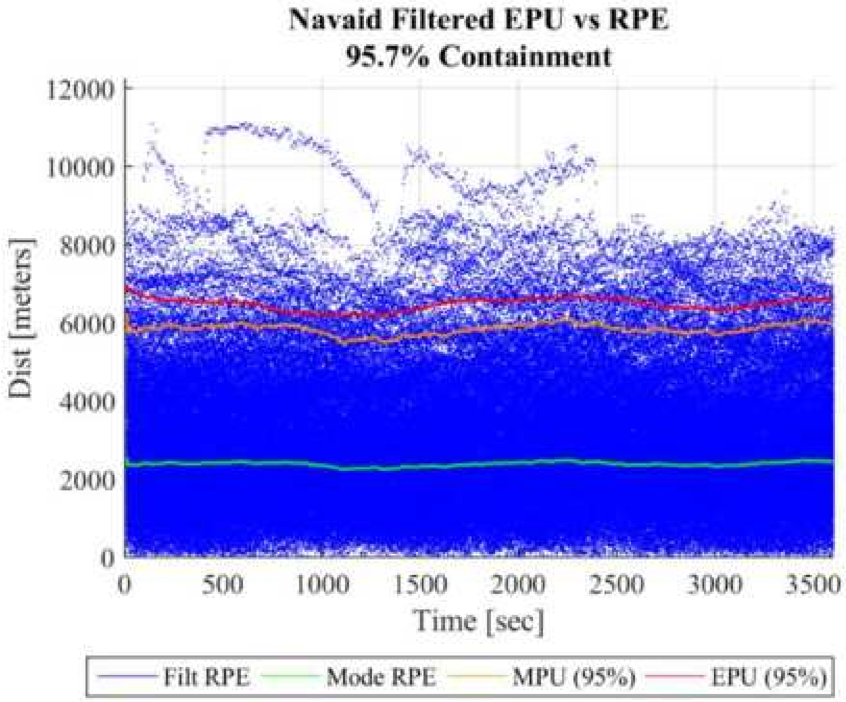

Figure J-14: Navigation Aid Filtered Lateral Position Accuracy for the 8 Navigation Aid, NACp 2, GVA 1, 0.7° AOA Uncertainty Test Case.

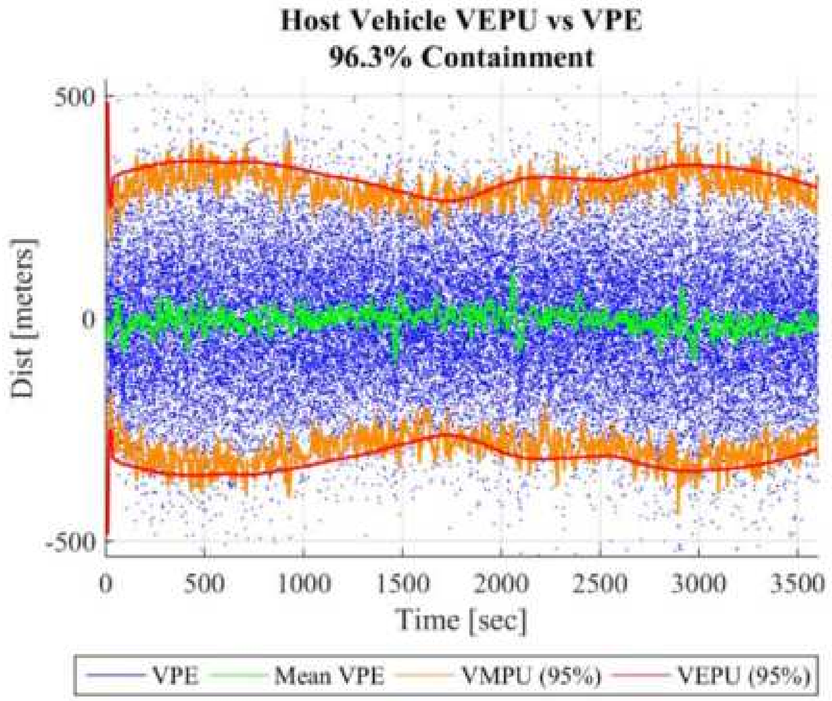

Figure J-15: Host Vehicle Vertical Position Accuracy for the 8 Navigation Aid, NACp 2, GVA 1, 0.7° AOE Uncertainty Test Case.

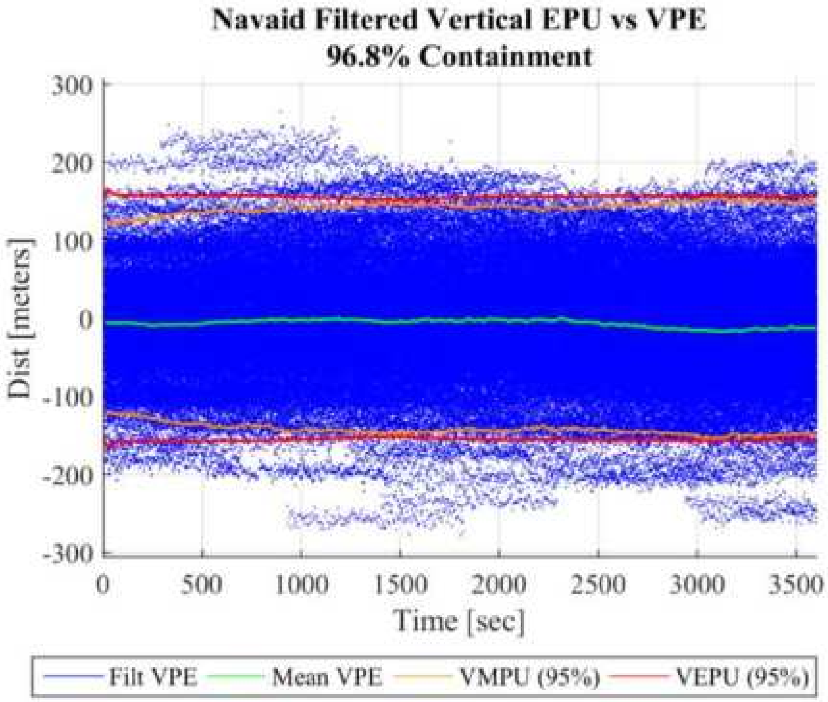

Figure J-16: Navigation Aid Filtered Vertical Position Accuracy for the 8 Navigation Aid, NACp 2, GVA 1, 0.7° AOA Uncertainty Test Case.

J.5 Parametric Test Case 5: 4 Navigation Aids, NACp 10, GVA 2, 0.7° AOA Uncertainty

The summary statistics for Parametric Test Case 5 are provided in Table J-5, with supporting data plots provided in Figure J-17 through Figure J-20.

Table J-5: Summary Statistics for Parametric Test Case 5.
4 Navigation Aids, NACp 10, GVA 2, 0.7° AOA/AOE Uncertainty.

Summary Statistic	Units	Value	Summary Statistic	Units	Value
Average Host EPU	[m]	1,045	Average Navaid EPU	[m]	73.1
Average Host RPE Mode	[m]	292.1	Average Navaid RPE Mode	[m]	24.2
Host EPU Containment	[%]	98.1	Navaid EPU Containment	[%]	96.9
Average Host VEPU	[m]	400.1	Average Navaid VEPU	[m]	52.9
Average Host VPE	[m]	6.1	Average Navaid VPE	[m]	-4.5
Host VEPU Containment	[%]	96.5	Navaid VEPU Containment	[%]	97.9

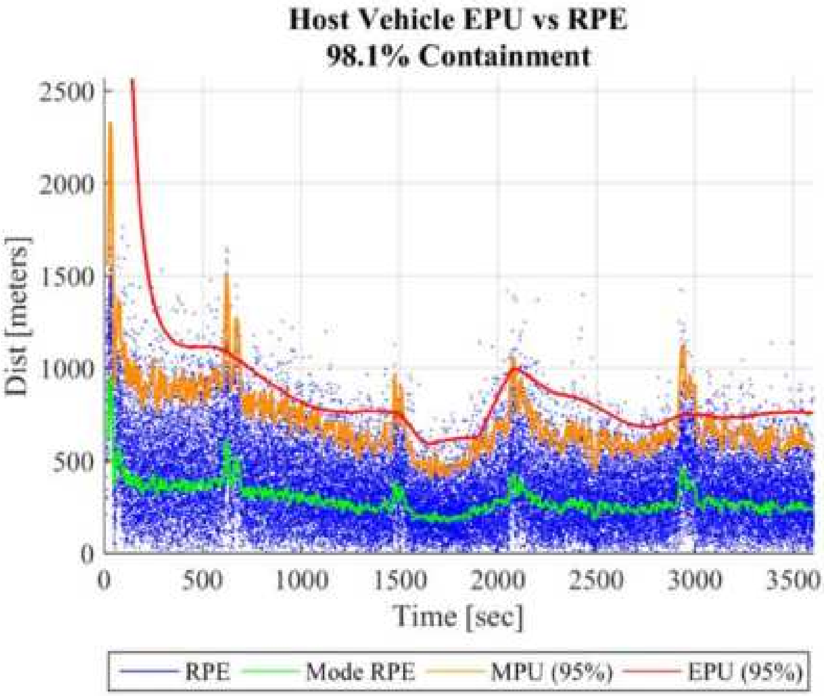

Figure J-17: Host Vehicle Lateral Position Accuracy for the 4 Navigation Aid, NACp 10, GVA 2, 0.7° AOA Uncertainty Test Case.

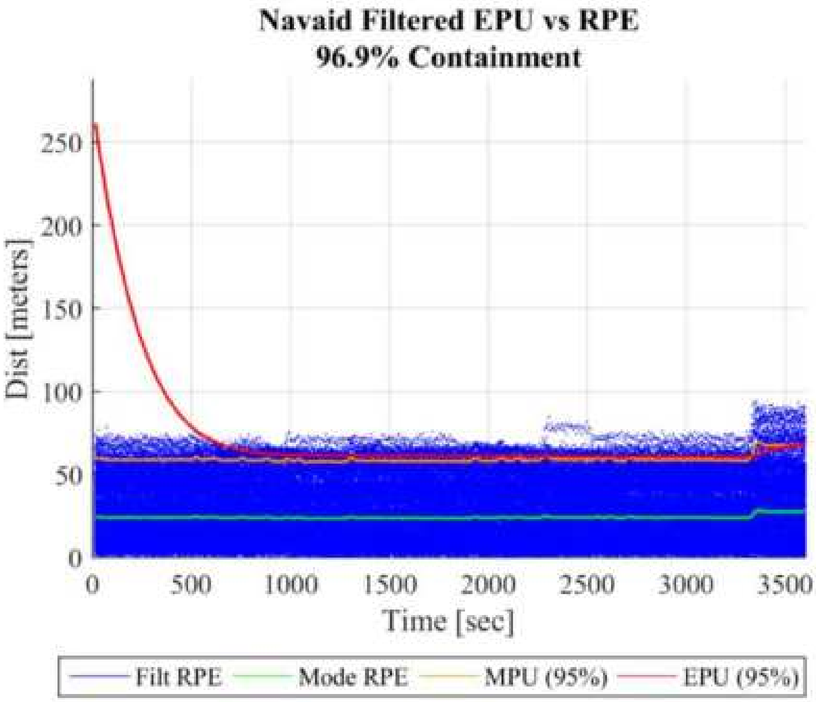

Figure J-18: Navigation Aid Filtered Lateral Position Accuracy for the 4 Navigation Aid, NACp 10, GVA 2, 0.7° AOA Uncertainty Test Case.

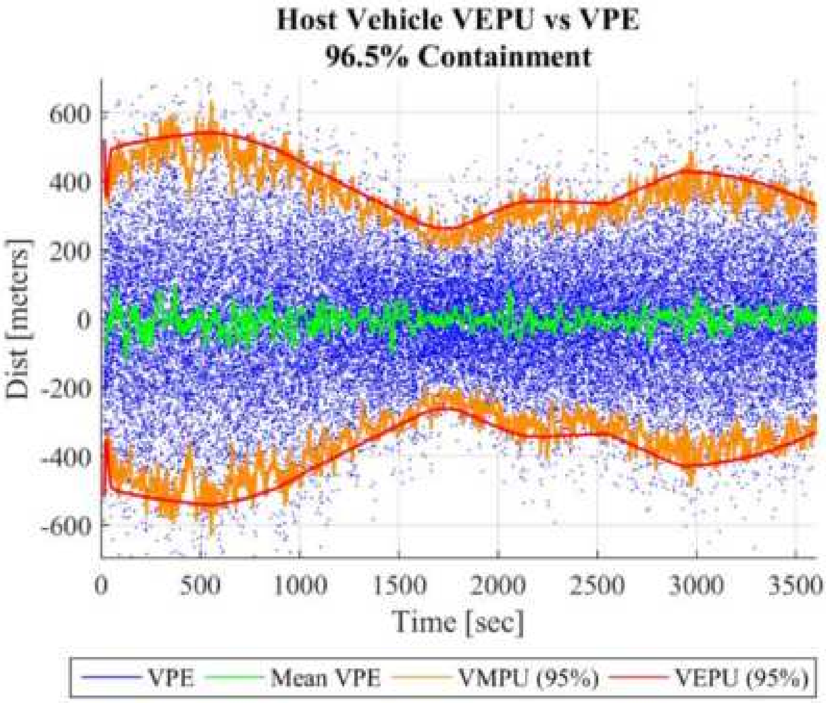

Figure J-19: Host Vehicle Vertical Position Accuracy for the 4 Navigation Aid, NACp 10, GVA 2, 0.7° AOE Uncertainty Test Case.

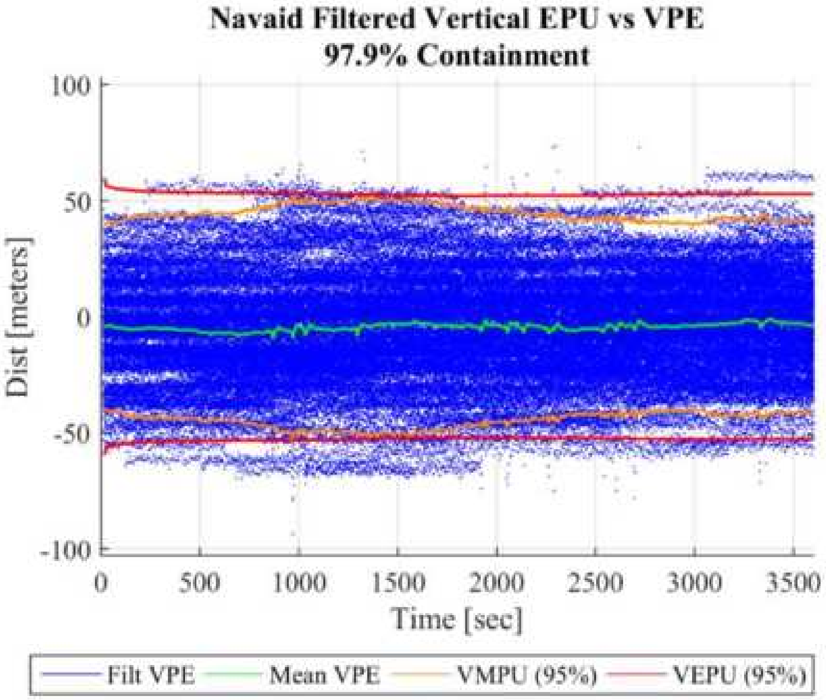

Figure J-20: Navigation Aid Filtered Vertical Position Accuracy for the 4 Navigation Aid, NACp 10, GVA 2, 0.7° AOA Uncertainty Test Case.

J.6 Parametric Test Case 6: 4 Navigation Aids, NACp 6, GVA 1, 0.7° AOA Uncertainty

The summary statistics for Parametric Test Case 6 are provided in Table J-6, with supporting data plots provided in Figure J-21 through Figure J-24.

Table J-6: Summary Statistics for Parametric Test Case 6.
4 Navigation Aids, NACp 6, GVA 1, 0.7° AOA/AOE Uncertainty.

Summary Statistic	Units	Value	Summary Statistic	Units	Value
Average Host EPU	[m]	1,193	Average Navaid EPU	[m]	590.7
Average Host RPE Mode	[m]	344.3	Average Navaid RPE Mode	[m]	221.5
Host EPU Containment	[%]	98.1	Navaid EPU Containment	[%]	96.7
Average Host VEPU	[m]	410.6	Average Navaid VEPU	[m]	151.7
Average Host VPE	[m]	-5.5	Average Navaid VPE	[m]	-6.6
Host VEPU Containment	[%]	96.6	Navaid VEPU Containment	[%]	96.1

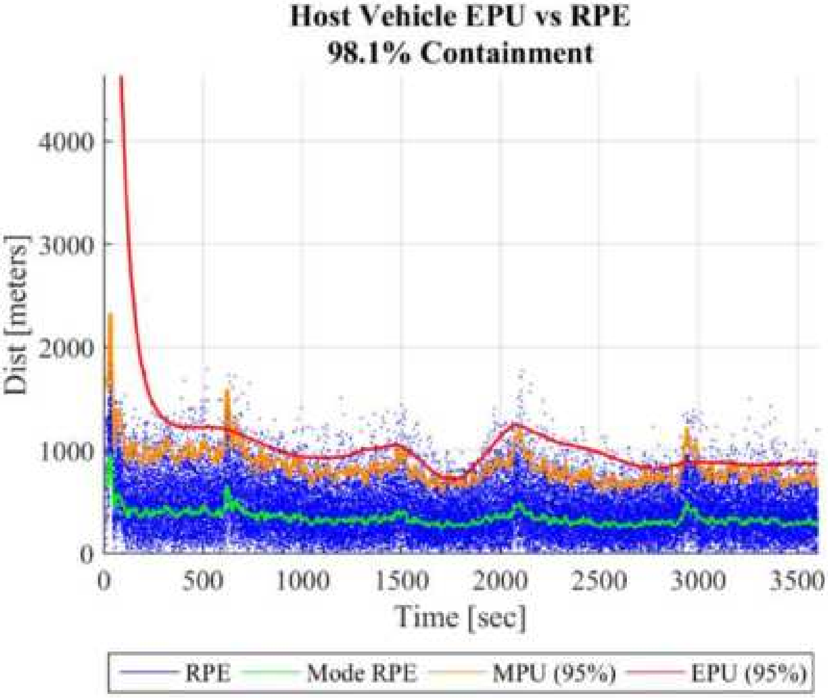

Figure J-21: Host Vehicle Lateral Position Accuracy for the 4 Navigation Aid, NACp 6, GVA 1, 0.7° AOA Uncertainty Test Case.

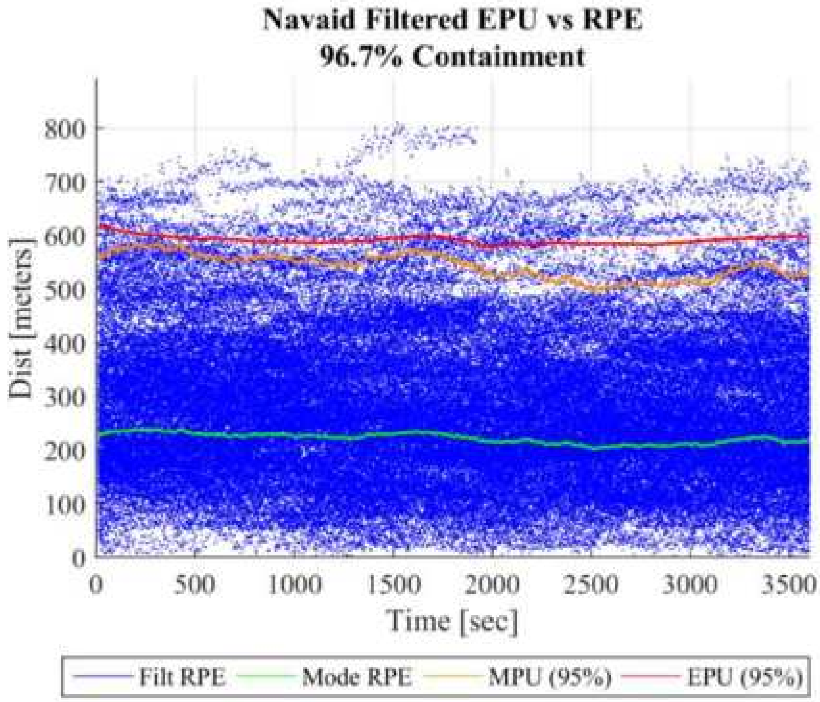

Figure J-22: Navigation Aid Filtered Lateral Position Accuracy for the 4 Navigation Aid, NACp 6, GVA 1, 0.7° AOA Uncertainty Test Case.

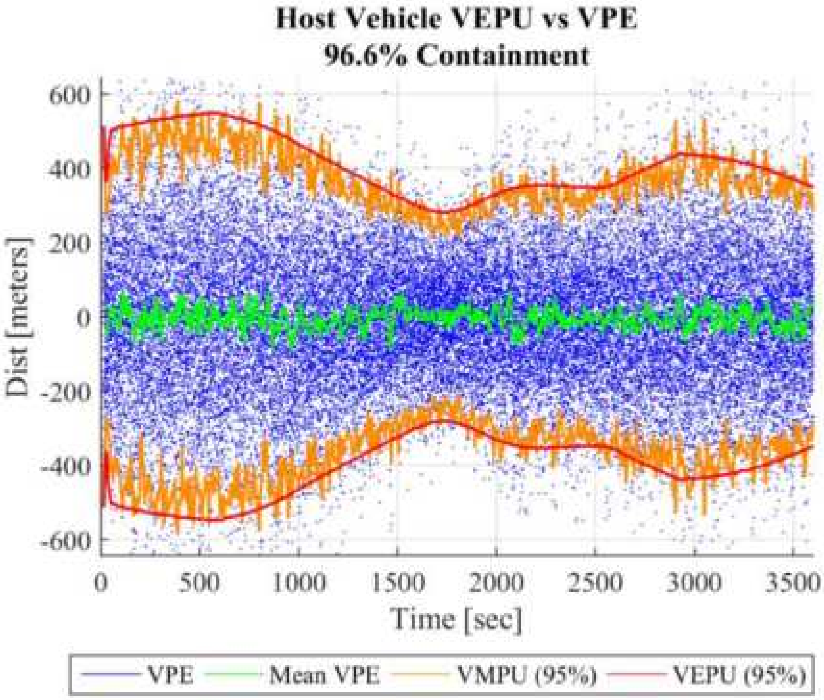

Figure J-23: Host Vehicle Vertical Position Accuracy for the 4 Navigation Aid, NACp 6, GVA 1, 0.7° AOE Uncertainty Test Case.

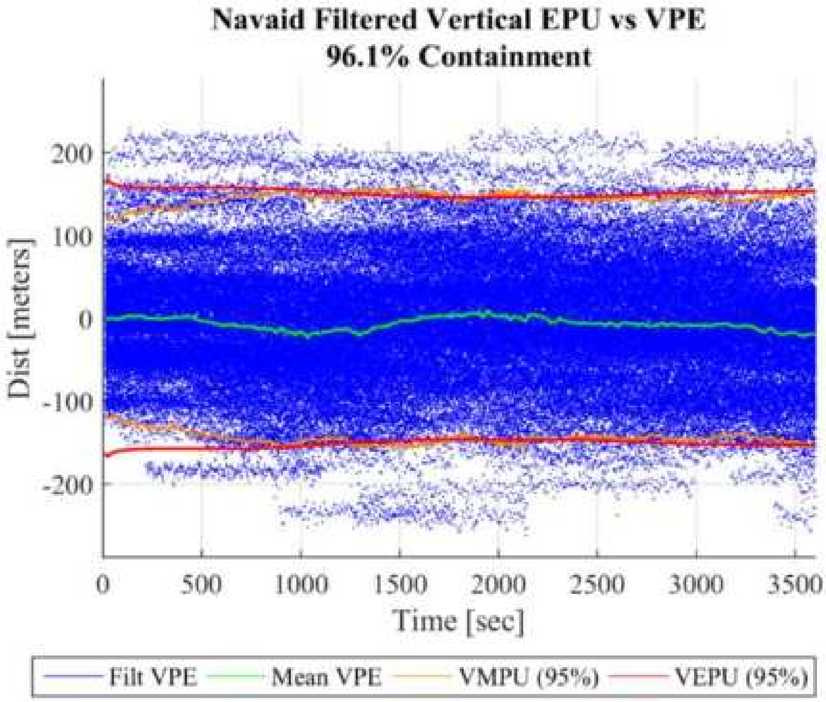

Figure J-24: Navigation Aid Filtered Vertical Position Accuracy for the 4 Navigation Aid, NACp 6, GVA 1, 0.7° AOA Uncertainty Test Case.

J.7 Parametric Test Case 7: 4 Navigation Aids, NACp 2, GVA 1, 0.7° AOA Uncertainty

The summary statistics for Parametric Test Case 7 are provided in Table J-8, with supporting data plots provided in Figure J-25 through Figure J-29.

Table J-7: Summary Statistics for Parametric Test Case 7.
4 Navigation Aids, NACp 2, GVA 1, 0.7° AOA/AOE Uncertainty.

Summary Statistic	Units	Value	Summary Statistic	Units	Value
Average Host EPU	[m]	6,058	Average Navaid EPU	[m]	6,898
Average Host RPE Mode	[m]	2,320	Average Navaid RPE Mode	[m]	2,510
Host EPU Containment	[%]	94.8	Navaid EPU Containment	[%]	96.5
Average Host VEPU	[m]	418.8	Average Navaid VEPU	[m]	157.6
Average Host VPE	[m]	-4.1	Average Navaid VPE	[m]	-4.4
Host VEPU Containment	[%]	96.4	Navaid VEPU Containment	[%]	97.6

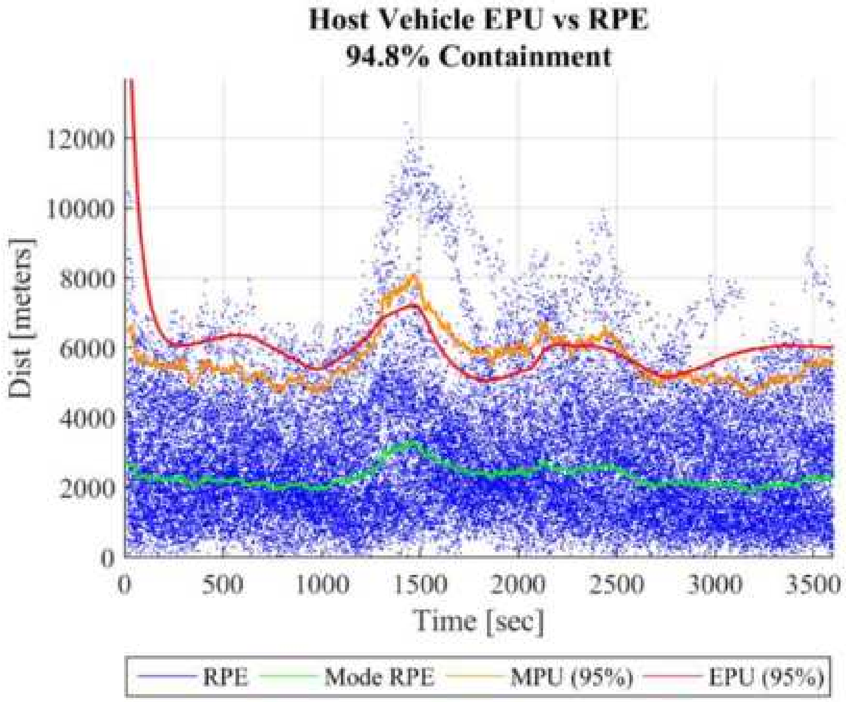

Figure J-25: Host Vehicle Lateral Position Accuracy for the 4 Navigation Aid, NACp 2, GVA 1, 0.7° AOA Uncertainty Test Case.

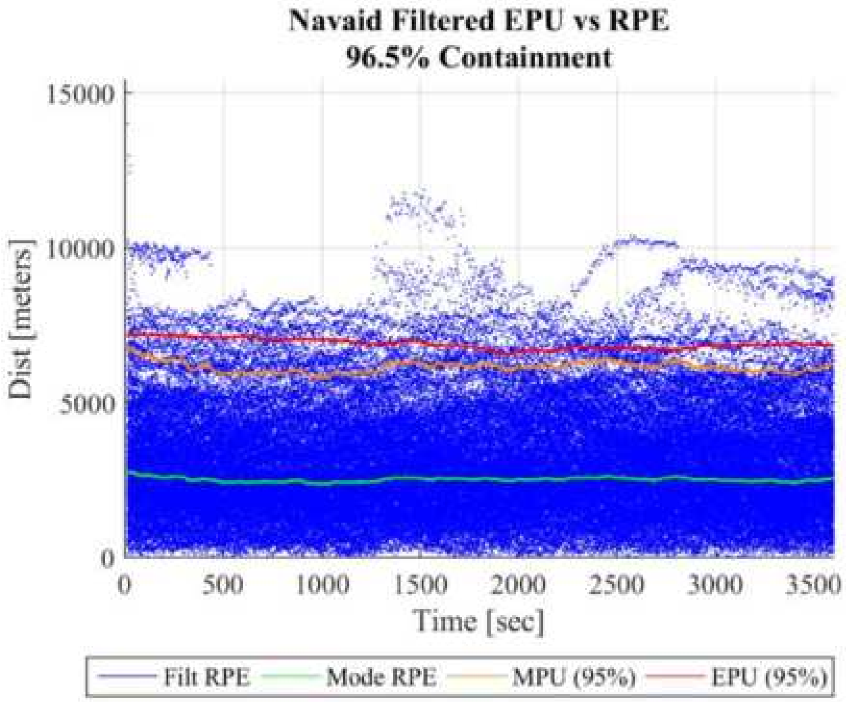

Figure J-26: Navigation Aid Filtered Lateral Position Accuracy for the 4 Navigation Aid, NACp 2, GVA 1, 0.7° AOA Uncertainty Test Case.

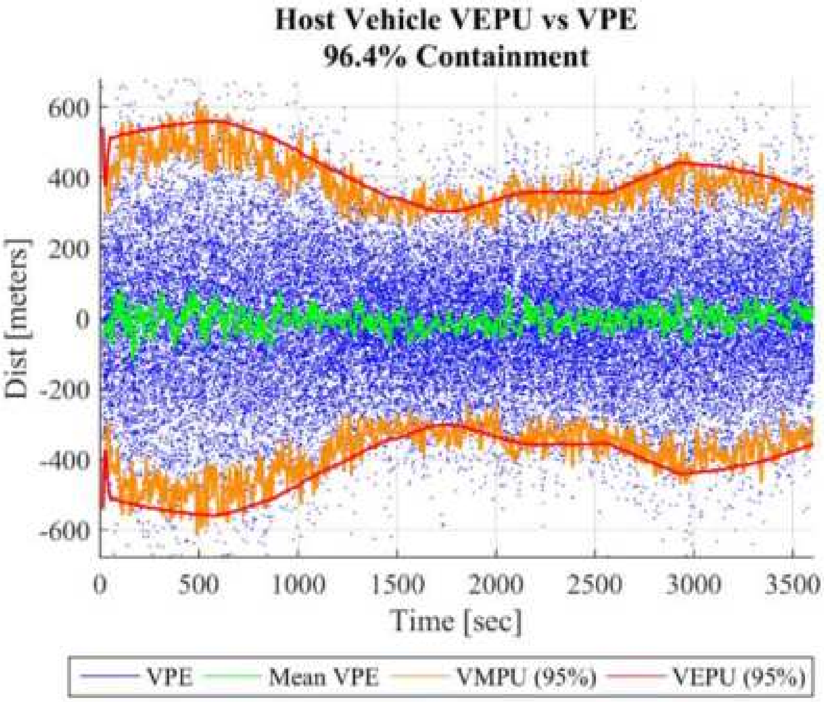

Figure J-27: Host Vehicle Vertical Position Accuracy for the 4 Navigation Aid, NACp 2, GVA 1, 0.7° AOE Uncertainty Test Case.

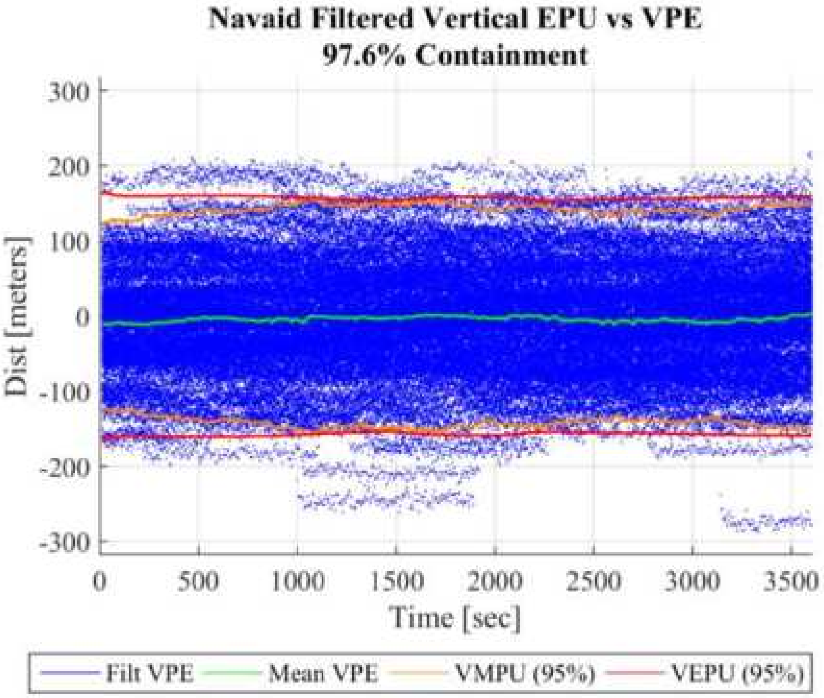

Figure J-28: Navigation Aid Filtered Vertical Position Accuracy for the 4 Navigation Aid, NACp 2, GVA 1, 0.7° AOA Uncertainty Test Case.

J.8 Parametric Test Case 8: 2 Navigation Aids, NACp 10, GVA 2, 0.7° AOA Uncertainty

The summary statistics for Parametric Test Case 8 are provided in Table J-8, with supporting data plots provided in Figure J-29 through Figure J-32.

Table J-8: Summary Statistics for Parametric Test Case 8.
2 Navigation Aids, NACp 10, GVA 2, 0.7° AOA/AOE Uncertainty.

Summary Statistic	Units	Value	Summary Statistic	Units	Value
Average Host EPU	[m]	2,031	Average Navaid EPU	[m]	69.2
Average Host RPE Mode	[m]	533.2	Average Navaid RPE Mode	[m]	23.5
Host EPU Containment	[%]	98.7	Navaid EPU Containment	[%]	97.0
Average Host VEPU	[m]	579.4	Average Navaid VEPU	[m]	53.4
Average Host VPE	[m]	-3.2	Average Navaid VPE	[m]	-2.4
Host VEPU Containment	[%]	96.5	Navaid VEPU Containment	[%]	97.1

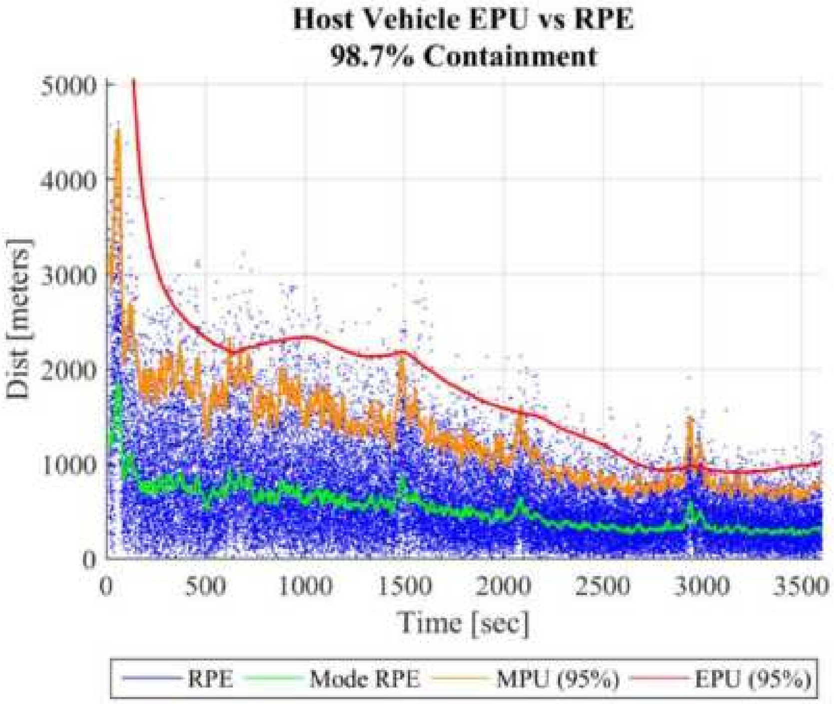

Figure J-29: Host Vehicle Lateral Position Accuracy for the 2 Navigation Aid, NACp 10, GVA 2, 0.7° AOA Uncertainty Test Case.

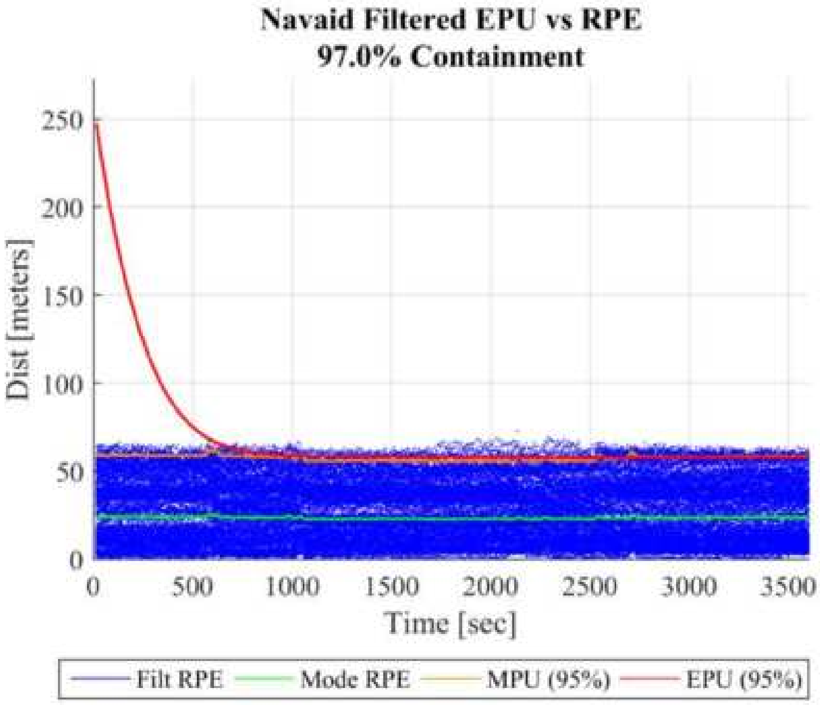

Figure J-30: Navigation Aid Filtered Lateral Position Accuracy for the 2 Navigation Aid, NACp 10, GVA 2, 0.7° AOA Uncertainty Test Case.

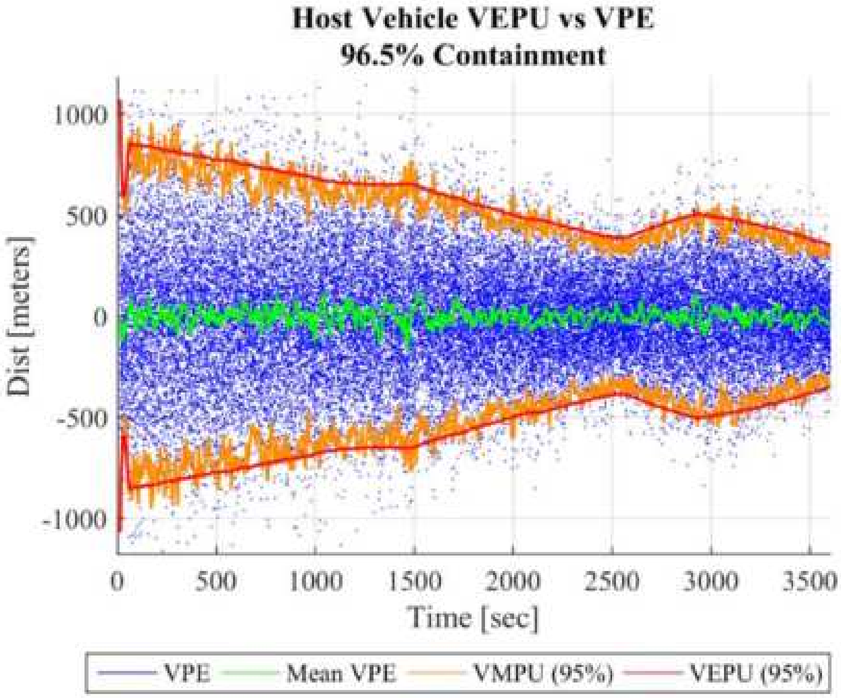

Figure J-31: Host Vehicle Vertical Position Accuracy for the 2 Navigation Aid, NACp 10, GVA 2, 0.7° AOE Uncertainty Test Case.

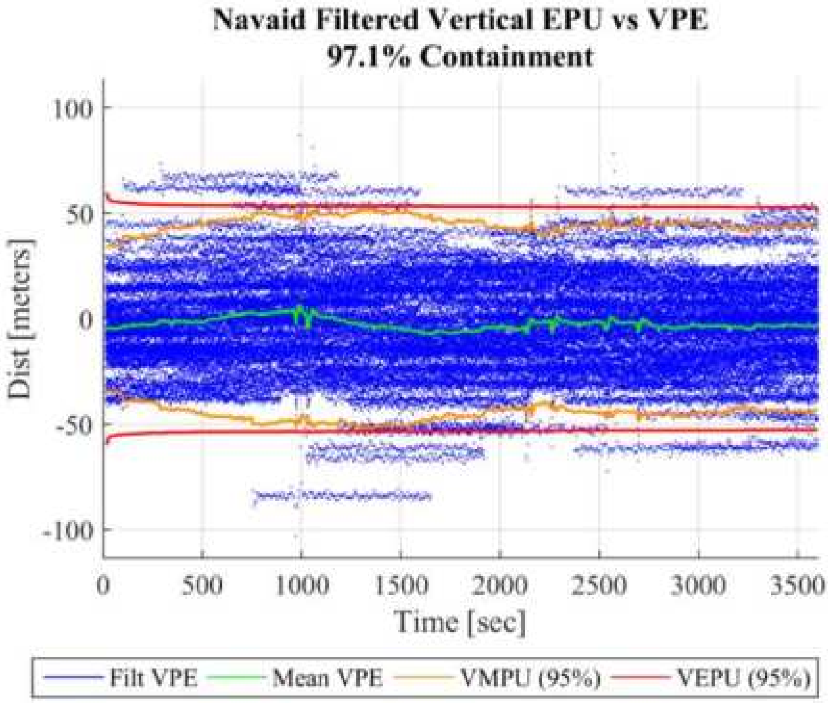

Figure J-32: Navigation Aid Filtered Vertical Position Accuracy for the 2 Navigation Aid, NACp 10, GVA 2, 0.7° AOA Uncertainty Test Case.

J.9 Parametric Test Case 9: 2 Navigation Aids, NACp 6, GVA 1, 0.7° AOA Uncertainty

The summary statistics for Parametric Test Case 9 are provided in Table J-9, with supporting data plots provided in Figure J-33 through Figure J-36.

Table J-9: Summary Statistics for Parametric Test Case 9.
2 Navigation Aids, NACp 6, GVA 1, 0.7° AOA/AOE Uncertainty.

Summary Statistic	Units	Value	Summary Statistic	Units	Value
Average Host EPU	[m]	2,343	Average Navaid EPU	[m]	611.6
Average Host RPE Mode	[m]	643.7	Average Navaid RPE Mode	[m]	235.6
Host EPU Containment	[%]	98.7	Navaid EPU Containment	[%]	97.3
Average Host VEPU	[m]	593.3	Average Navaid VEPU	[m]	159.8
Average Host VPE	[m]	-5.2	Average Navaid VPE	[m]	-5.4
Host VEPU Containment	[%]	96.7	Navaid VEPU Containment	[%]	96.7

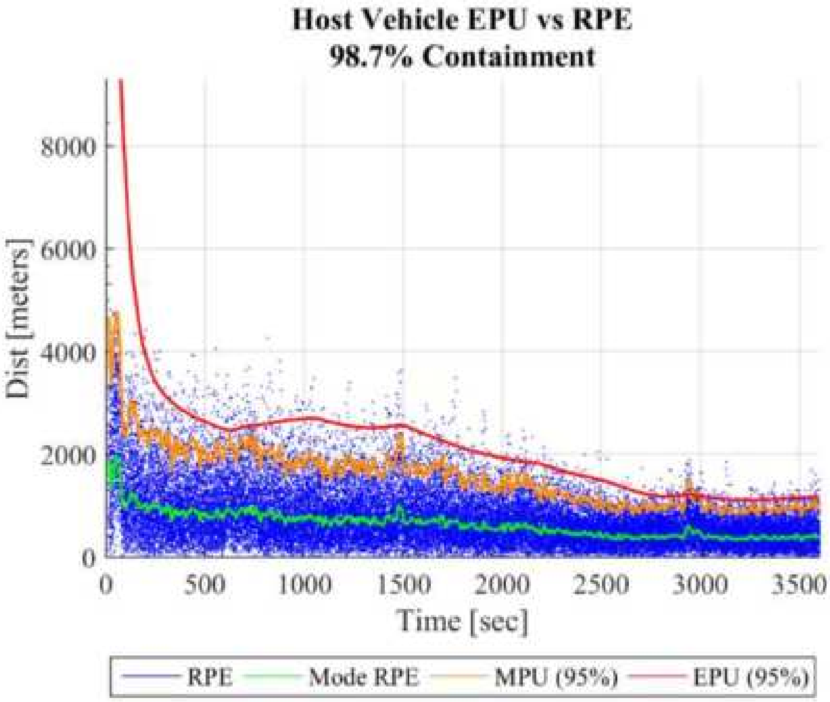

Figure J-33: Host Vehicle Lateral Position Accuracy for the 2 Navigation Aid, NACp 6, GVA 1, 0.7° AOA Uncertainty Test Case.

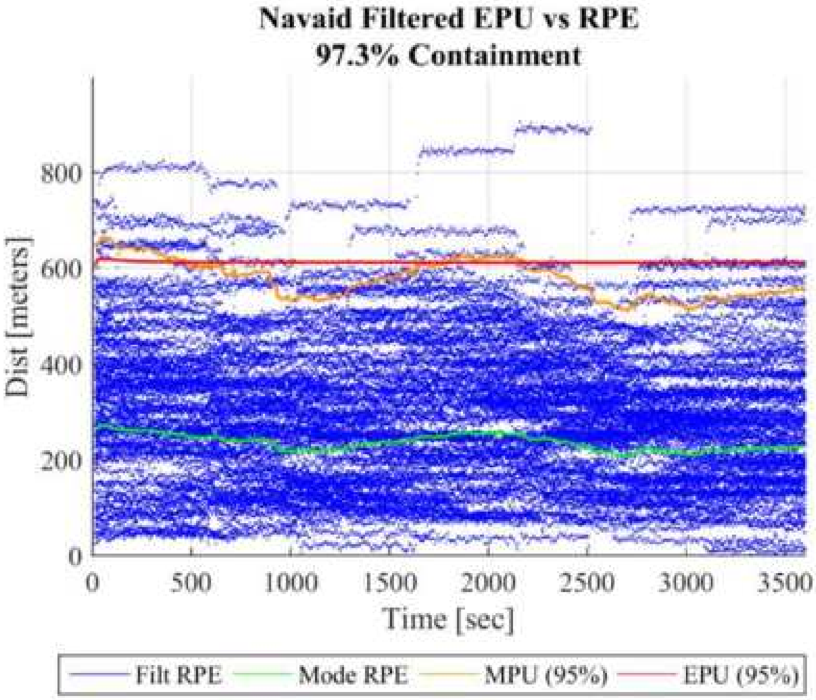

Figure J-34: Navigation Aid Filtered Lateral Position Accuracy for the 2 Navigation Aid, NACp 6, GVA 1, 0.7° AOA Uncertainty Test Case.

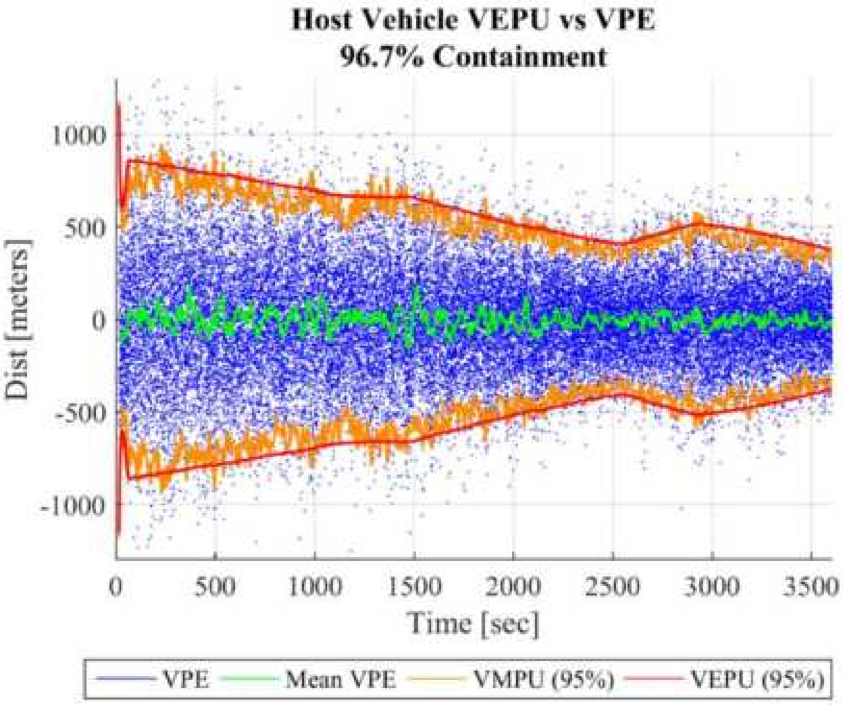

Figure J-35: Host Vehicle Vertical Position Accuracy for the 2 Navigation Aid, NACp 6, GVA 1, 0.7° AOE Uncertainty Test Case.

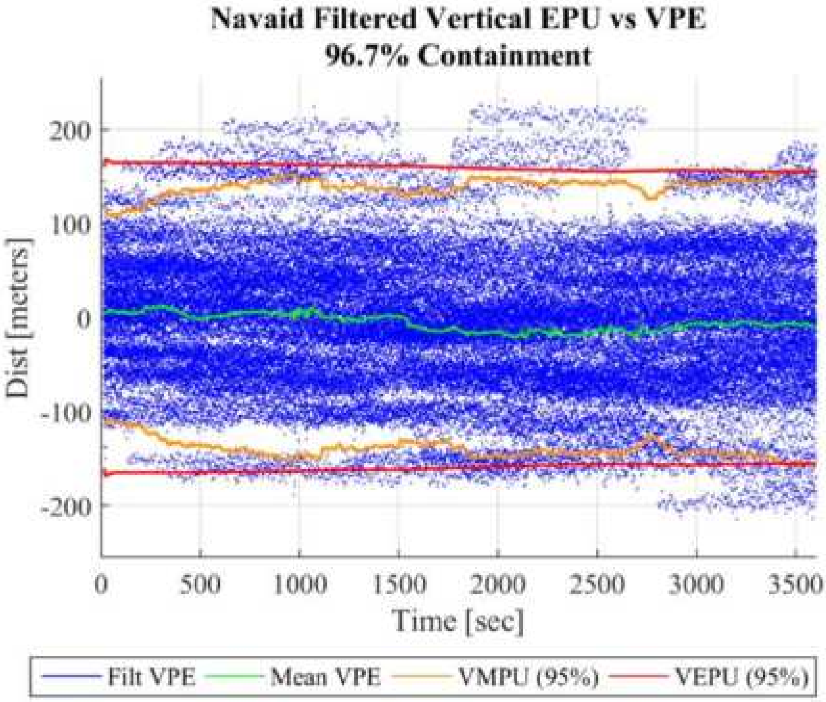

Figure J-36: Navigation Aid Filtered Vertical Position Accuracy for the 2 Navigation Aid, NACp 6, GVA 1, 0.7° AOA Uncertainty Test Case.

J.10 Parametric Test Case 10: 2 Navigation Aids, NACp 2, GVA 1, 0.7° AOA Uncertainty

The summary statistics for Parametric Test Case 10 are provided in Table J-10, with supporting data plots provided in Figure J-37 through Figure J-40.

Table J-10: Summary Statistics for Parametric Test Case 10.
2 Navigation Aids, NACp 2, GVA 1, 0.7° AOA/AOE Uncertainty.

Summary Statistic	Units	Value	Summary Statistic	Units	Value
Average Host EPU	[m]	13,090	Average Navaid EPU	[m]	7,470
Average Host RPE Mode	[m]	4,199	Average Navaid RPE Mode	[m]	2,999
Host EPU Containment	[%]	98.4	Navaid EPU Containment	[%]	96.1
Average Host VEPU	[m]	681.5	Average Navaid VEPU	[m]	162.9
Average Host VPE	[m]	-6.2	Average Navaid VPE	[m]	-3.5
Host VEPU Containment	[%]	98.1	Navaid VEPU Containment	[%]	96.4

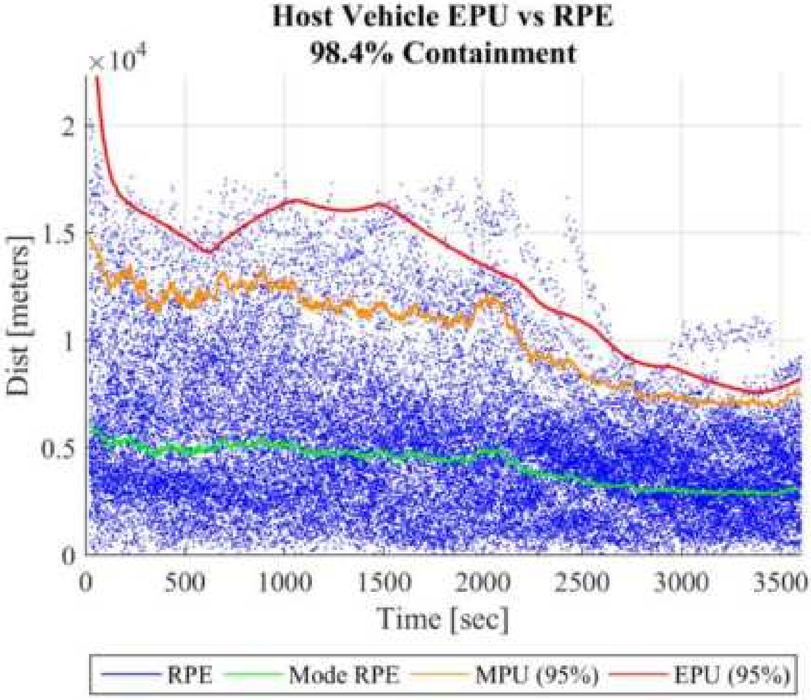

Figure J-37: Host Vehicle Lateral Position Accuracy for the 2 Navigation Aid, NACp 2, GVA 1, 0.7° AOA Uncertainty Test Case.

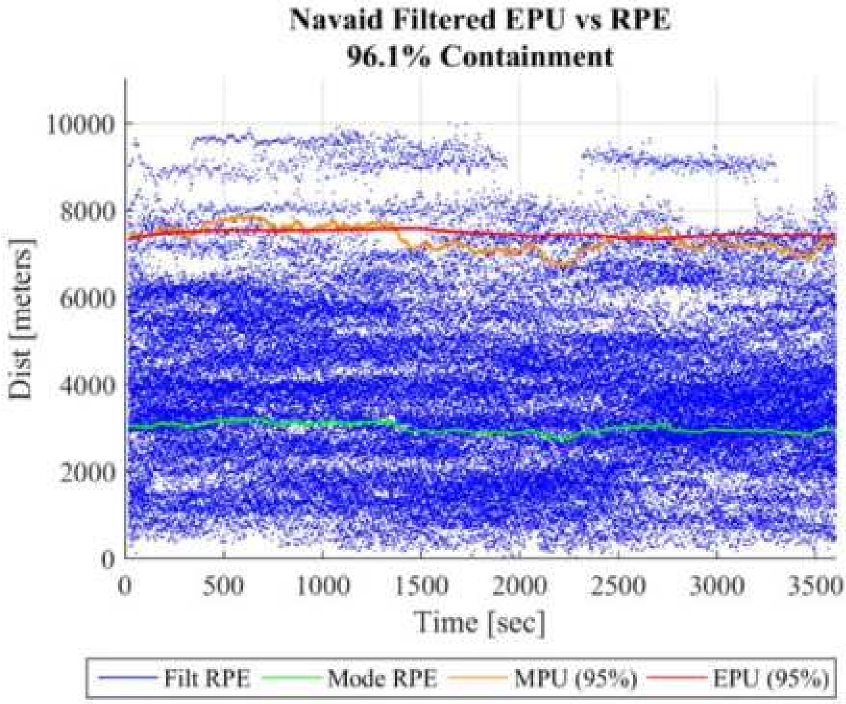

Figure J-38: Navigation Aid Filtered Lateral Position Accuracy for the 2 Navigation Aid, NACp 2, GVA 1, 0.7° AOA Uncertainty Test Case.

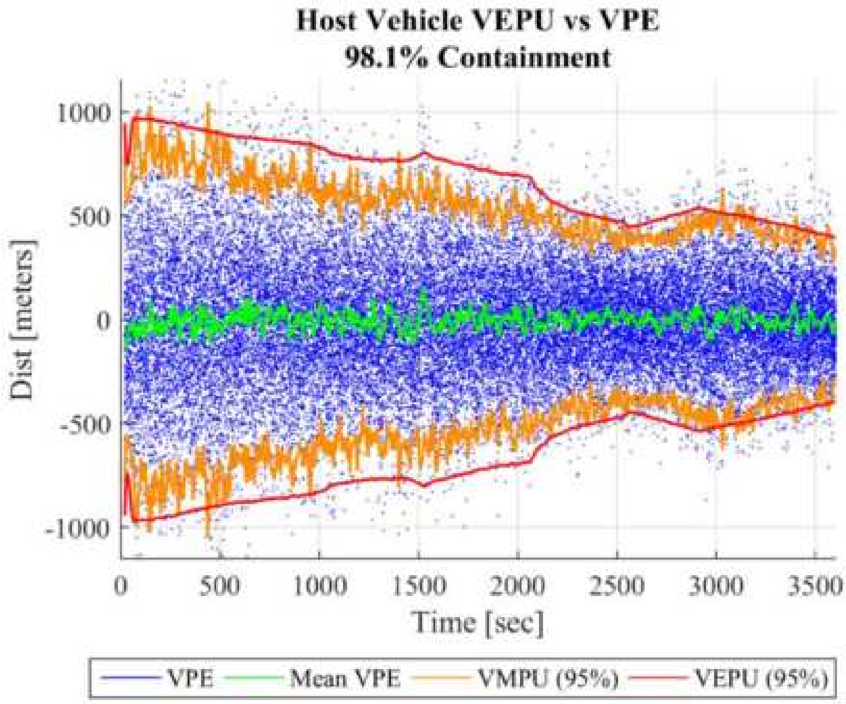

Figure J-39: Host Vehicle Vertical Position Accuracy for the 2 Navigation Aid, NACp 2, GVA 1, 0.7° AOE Uncertainty Test Case.

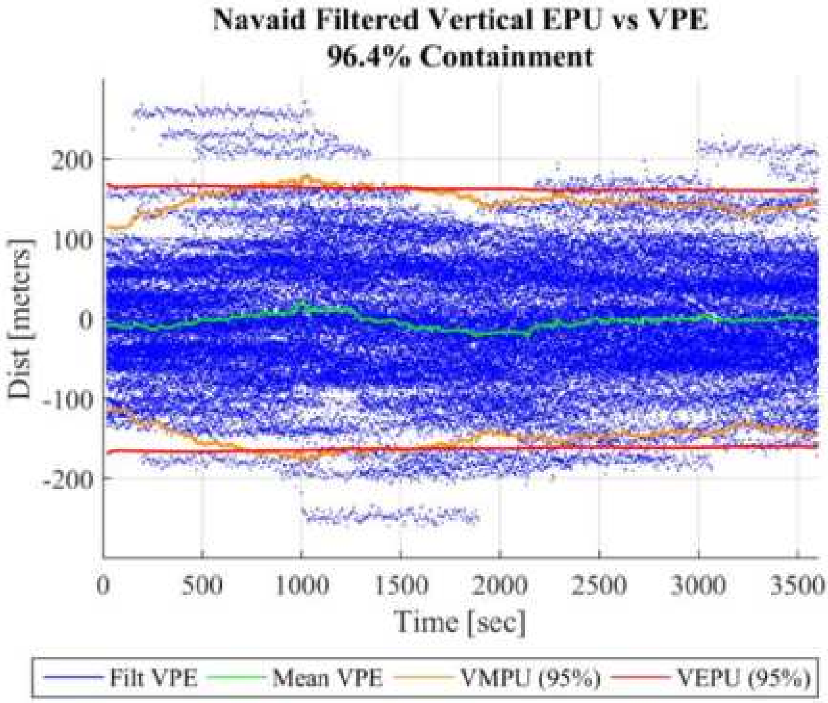

Figure J-40: Navigation Aid Filtered Vertical Position Accuracy for the 2 Navigation Aid, NACp 2, GVA 1, 0.7° AOA Uncertainty Test Case.

REFERENCES

[1] RTCA, Inc., “Minimum operational performance standards for 1090 MHz extended squitter automatic dependent surveillance - broadcast and traffic information services - broadcast,” Washington, DC, DO-260B, Dec. 2011.

[2] C. Reck, U. Berold, J. Weinzier, and L. P. Schmidt, “Direction of arrival estimation from secondary surveillance radar signals in presence of hardware imperfections,” in *Radar Conference, 2008. EuRAD 2008. European*, Amsterdam, 2008, pp. 252–255 [Online]. Available: http://ieeexplore.ieee.org.ezproxy.gvsu.edu/stamp/stamp.jsp?tp=&arnumber=4760849&isnumber=4760754. [Accessed: 12-Sep-2016]

[3] C. Reck, U. Berold, J. Schur, and L. P. Schmidt, “Direction of arrival sensor calibration based on ADS-B airborne position telegrams,” in *Radar Conference, 2009. EuRAD 2009. European*, Rome, 2009, pp. 77–80 [Online]. Available: http://ieeexplore.ieee.org.ezproxy.gvsu.edu/stamp/stamp.jsp?tp=&arnumber=5307165&isnumber=5306978. [Accessed: 12-Sep-2016]

[4] C. Reck, U. Berold, and L. P. Schmidt, “High precision DOA estimation of SSR transponder signals,” in *Wireless Information Technology and Systems (ICWITS), 2010 IEEE International Conference on*, Honolulu, HI, 2010, pp. 1–4 [Online]. Available: http://ieeexplore.ieee.org.ezproxy.gvsu.edu/stamp/stamp.jsp?tp=&arnumber=5611975&isnumber=5611808. [Accessed: 12-Sep-2016]

[5] C. Reck, U. Berold, and L. P. Schmidt, “Robust DOA estimation of SSR signals for aircraft positioning,” in *Wireless Sensors and Sensor Networks (WiSNet), 2011 IEEE Topical Conference on*, Phoenix, AZ, 2011, pp. 13–16 [Online]. Available: http://ieeexplore.ieee.org.ezproxy.gvsu.edu/stamp/stamp.jsp?tp=&arnumber=5725024&isnumber=5725016. [Accessed: 12-Sep-2016]

[6] C. Reck, M. S. Reuther, A. Jasch, and L. P. Schmidt, “Independent surveillance broadcast — ADS-B receivers with DOA estimation,” in *Digital Communications - Enhanced Surveillance of Aircraft and Vehicles (TIWDC/ESAV), 2011 Tyrrhenian International Workshop on*, Capri, 2011, pp. 219–222 [Online]. Available: http://ieeexplore.ieee.org.ezproxy.gvsu.edu/stamp/stamp.jsp?tp=&arnumber=6060992&isnumber=6060948. [Accessed: 12-Sep-2016]

[7] C. Reck, M. S. Reuther, A. Jasch, and L.-P. Schmidt, “Verification of ADS-B positioning by direction of arrival estimation,” *International Journal of Microwave and Wireless Technologies*, vol. 4, no. 2, pp. 181–186, Feb. 2012.

[8] R. Faragher, P. MacDoran, and M. Mathews, “Spoofing mitigation, robust collision avoidance, and opportunistic receiver localisation using a new signal processing scheme for ADS-B or AIS,” in *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014)*, Tampa, FL, 2014, pp. 858–868.

[9] "Applied signals intelligence." [Online]. Available: www.asigint.com. [Accessed: 02-Jun-2017]

[10] "Aaronia AG." [Online]. Available: www.aaronia.com. [Accessed: 02-Jun-2017]

[11] Federal Aviation Administration, Department of Transportation, "Airworthiness approval of automatic dependent surveillance - broadcast out systems," Advisory Circular AC 20-165B, Dec. 2015 [Online]. Available: http://www.faa.gov/documentLibrary/media/Advisory_Circular/AC_20-165B.pdf. [Accessed: 12-Jul-2016]

[12] Federal Aviation Administration, Department of Transportation, "Automatic dependent surveillance - broadcast operations," Advisory Circular AC 90-114A, Oct. 2014 [Online]. Available: http://www.faa.gov/documentLibrary/media/Advisory_Circular/AC_90-114A_FAA_Web_(2).pdf. [Accessed: 12-Jul-2016]

[13] U.S. Department of Transportation - Bureau of Transportation Statistics, "National transportation statistics," Jul. 2017 [Online]. Available: https://www.rita.dot.gov/bts/sites/rita.dot.gov.bts/files/publications/national_transportation_statistics/index.html. [Accessed: 17-Oct-2017]

[14] A. Helfrick, *Principles of avionics*, 9th ed. Leesburg, VA: Avionics Communications Inc., 2015.

[15] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 7, pp. 984–995, Jul. 1989.

[16] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, Mar. 1986.

[17] M. Haardt and F. Romer, "Enhancements of unitary ESPRIT for non-circular sources," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, 2004, pp. ii-101–4 vol.2 [Online]. Available: http://ieeexplore.ieee.org.ezproxy.gvsu.edu/stamp/stamp.jsp?tp=&arnumber=1326204&isnumber=29344. [Accessed: 12-Sep-2016]

[18] M. Haardt and M. E. Ali-Hackl, "Unitary ESPRIT: how to exploit additional information inherent in the relational invariance structure," in *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, Adelaide, SA, 1994, vol. 4, p. IV/229-IV/232 [Online]. Available: http://ieeexplore.ieee.org.ezproxy.gvsu.edu/stamp/stamp.jsp?tp=&arnumber=389832&isnumber=8836. [Accessed: 15-Sep-2016]

[19] Y. Hua and T. K. Sarkar, "Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 5, pp. 814–824, May 1990.

[20] G. Welch and G. Bishop, “An introduction to the Kalman Filter,” University of North Carolina at Chapel Hill, Chapel Hill, NC, Technical Report 95–041, Jul. 2006 [Online]. Available: https://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf. [Accessed: 14-Jul-2016]

[21] C. K. Chui and G. Chen, *Kalman filtering with real-time applications*, 4th ed. Berlin Heidelberg: Springer-Verlag, 2009.

[22] M. Grewal and A. Andrews, *Kalman filtering - theory and practice using MATLAB*. Hoboken, NJ: Wiley, 2008.

[23] C.-B. Chang and K.-P. Dunn, *Applied state estimation and association*. Cambridge, MA: MIT Press, 2016.

[24] P. Kim, *Kalman filter for beginners with MATLAB examples*. Korea: A-JIN, 2011.

[25] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part I,” *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[26] B. Sicillana and O. Khatib, Eds., *Springer handbook of robotics*. Berlin Heidelberg: Springer, 2008 [Online]. Available: http://link.springer.com.ezproxy.gvsu.edu/book/10.1007%2F978-3-540-30301-5. [Accessed: 22-Jul-2016]

[27] S. Huang and G. Dissanayake, “Convergence and consistency analysis for extended Kalman filter based SLAM,” *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1036–1049, Oct. 2007.

[28] J. Shynk, *Probability, random variables, and random processes : Theory and signal processing applications*. Hoboken, New Jersey: Wiley-Interscience, 2012 [Online]. Available: http://site.ebrary.com.ezproxy.gvsu.edu/lib/gvsu/reader.action?docID=10605317. [Accessed: 08-Jul-2017]

[29] R. Yates and D. Goodman, *Probability and stochastic processes*, 2nd ed. Wiley, 2004.

[30] C. O. Archer, “Some properties of Rayleigh distributed random variables and of their sums and products,” Naval Missile Center, Point Mugu, CA, Technical Memorandum TM-67-15, Apr. 1967 [Online]. Available: https://apps.dtic.mil/dtic/tr/fulltext/u2/650090.pdf. [Accessed: 14-Jan-2019]

[31] F. van Diggelen, “GNSS accuracy: lies, damn lies, and statistics,” *GPS World*, vol. 18, no. 1, Jan-2007 [Online]. Available: http://gpsworld.com/gpsgnss-accuracy-lies-damn-lies-and-statistics-1134/. [Accessed: 04-Aug-2017]

[32] P. Groves, *Principles of GNSS, inertial, and multisensor integrated navigation systems*. London: Artech House, 2013.

[33] W. M. Smart, *Text-book on spherical astronomy*. Cambridge University Press, 1949.

[34] J. Iliffe and R. Lott, *Datums and map projections for remote sensing, GIS, and surveying*, 2nd ed. Scotland, United Kingdom: Whittles Publishing, 2008.

[35] M. Geyer, “Geometric analysis of an observer on a spherical Earth and an aircraft or satellite,” U.S. Department of Transportation - Federal Aviation Administration, DOT-VNTSC-FAA-13-08, Sep. 2013 [Online]. Available: http://ntl.bts.gov/lib/48000/48500/48549/Project_Memo_DOT-VNTSC-FAA-13-08.pdf. [Accessed: 08-Aug-2016]

[36] Federal Aviation Administration, Department of Transportation, “United States standard for performance based navigation instrument procedure design,” National Policy 8260.58A CHG 1, Mar. 2017 [Online]. Available: https://www.faa.gov/documentLibrary/media/Order/Order_8260_58A_CHG_1_Consolidated.pdf. [Accessed: 27-Aug-2017]

[37] T. Vincenty, “Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations,” *Survey Review*, vol. XXIII, no. 176, pp. 88–93, Apr. 1975.

[38] C. F. F. Karney, “Algorithms for geodesics,” *Journal of Geodesy*, vol. 87, no. 1, pp. 43–55, 2013.

[39] C. Ancker, “Airborne direction finding - the theory of navigation errors,” *IRE Transactions on Aeronautical and Navigational Electronics*, vol. ANE-5, no. 4, pp. 199–210, Dec. 1958.

[40] M. Kayton and W. Fried, *Avionics navigation systems*, 2nd ed. New York, NY: Wiley, 1997.

[41] A. Leick, L. Rapoport, and D. Tatarnikov, *GPS satellite surveying*, 4th ed. Hoboken, NJ: Wiley, 2015.

[42] A. G. Dempster, “Dilution of precision in angle-of-arrival positioning systems,” *Electronics Letters*, vol. 42, no. 5, Feb. 2006 [Online]. Available: http://ieeexplore.ieee.org.ezproxy.gvsu.edu/stamp/stamp.jsp?tp=&arnumber=1604871. [Accessed: 03-Jul-2017]

[43] T. Murphy and W. Harris, “Device, system and methods using angle of arrival measurements for ADS-B authentication and navigation,” U.S. Patent 2014/0327581 A1,06-Nov-2014.

[44] S. Anderson and A. Persson, “Validity check of vehicle position information,” U.S. Patent 2011/0163908 A1,07-Jul-2011.

[45] W. Hall, “System and method for ensuring ADS-B integrity of departing aircraft,” U.S. Patent 2015/9116240 B2,25-Aug-2015.

[46] J. Kim and H. Hmam, “3D self-localization from angle of arrival measurements,” Australian Government - Department of Defence - Defence Science and Technology

Organization, Edinburgh South Australia, Technical Report DSTO-TR-2278, Apr. 2009 [Online]. Available: http://www.dtic.mil/dtic/tr/fulltext/u2/a501746.pdf. [Accessed: 28-Oct-2017]

[47] G. Dissanayake, H. Durrant-Whyte, and T. Bailey, “A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem,” in *Robotics and Automation, 2000. Proceedings. ICRA ’00. IEEE International Conference on*, San Francisco, CA, 2000, vol. 2, pp. 1009–1014 [Online]. Available: http://ieeexplore.ieee.org.ezproxy.gvsu.edu/stamp/stamp.jsp?arnumber=844732. [Accessed: 01-Oct-2016]

[48] A. Gelb, Ed., *Applied optimal estimation*. Cambridge, MA: MIT Press, 1974.

[49] R. Labbe, “Kalman and Baysian filters in Python,” 15-Jul-2017. [Online]. Available: https://drive.google.com/file/d/0By_SW19c1BfhSVFzNHc0SjduNzg/view. [Accessed: 13-Aug-2017]

[50] C. Veness, “Vector-based geodesy,” 2016. [Online]. Available: http://www.movable-type.co.uk/scripts/latlong-vectors.html. [Accessed: 20-Aug-2016]

[51] R. Sinnott, “Virtues of Haversine,” *Sky & Telescope*, vol. 68, p. 1, Aug. 1984.