On the Treatment of Noise and Conspiring Bias in Dual-Frequency Differential Global Navigation Satellite Systems

A dissertation presented to

the faculty of

the Russ College of Engineering and Technology of Ohio University

In partial fulfillment

of the requirements for the degree

Doctor of Philosophy

Dean C. Bruckner

March 2010

© 2010 Dean C. Bruckner. All Rights Reserved.
This dissertation titled
On the Treatment of Noise and Conspiring Bias in Dual-Frequency Differential Global
Navigation Satellite Systems

by
DEAN C. BRUCKNER

has been approved for
the School of Electrical Engineering and Computer Science
and the Russ College of Engineering and Technology by

__________________________

Frank van Graas
Fritz J. and Dolores H. Russ Professor of Electrical Engineering

__________________________

Dennis Irwin
Dean, Russ College of Engineering and Technology

Rev. 31 Dec 2010; DCB

• CLARIFICATION: Throughout this document, the random variable (RV) used to model pseudorange error of a given GPS satellite is said to be uncorrelated to the RVs for pseudorange errors of other satellites. This assumption is accurate for the noise propagation models addressed in this document, since after the exchange of noise for bias in the Code Noise and Multipath (CNMP) algorithm the only noise term remaining is thermal noise. By definition, thermal noise is Gaussian and independent, which implies there is no correlation between RVs. The values of correlation matrix elements reflect this uncorrelated assumption, and the term “uncorrelated” used in this dissertation should be understood as applying to the noise models. When forming the protection level, however, the range domain bias errors of an RV are intentionally and perfectly correlated with those of the other RVs, so as to produce the worst case position domain bias. Since the noise and bias are treated separately using two different probability distributions—one Gaussian and the other a degenerate distribution consisting of a single maximum possible value—this is permissible. Put another way, the RVs that model the pseudoranges are indeed correlated, but only in their mean values that conspire as biases in the position domain. The modeling of noise propagation is unaffected.

• Page 30: Figure 3, block reading “Receive differential corrections and PR & CP measurements” should read, “Receive differential corrections, noise sigmas, bias bounds, PR & CP measurements.”

• Pages 75 – 77: Throughout subsection “Statistical Method for Exact VPLc and HPLc” change variable represented by Roman letter B to Greek letter β, including in Equation (31) and Fig. 5.

• Page 95: Equation (45) should read as follows:

\[ RPL_{c1} = K_{\text{final}}d_{\text{major}} + \max \left\{ \frac{1}{N} \sum \left( s_{1,j} + s_{2,j} \right) \right\}, \quad j = 1, 2, \ldots, L \]

• Page 96: Table 3, column 2, row 4 should contain revised Equation (45) shown above.

• Page 97: Table 4, row 6 should read as follows:

| Maximum range domain bias, \(B_{\text{max}}\) | 20 cm | Applied equally to all satellites as an upper bound. Includes group delay; does not include multipath or residual iono or tropo errors. |

Errata p. 1
• Page 100: Fig 12 should appear as follows:

Figure 12: Simulation NSE (noise and bias) for east, north and up axes (unsmoothed; 100,000 points)

• Page 104: Sentence “The same bias combination was selected by both methods, so RPLc values are the same for both” should read, “The same bias combination was selected by both methods, but the RPLc1 value is slightly larger than the exact value since noise and bias are arranged collinearly.”

• Page 104: Table 7 should appear as follows:

<table>
<thead>
<tr>
<th>RPLc Method</th>
<th>Value in m (99.99%)</th>
<th>Percent of simulated errors enclosed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact</td>
<td>4.732</td>
<td>99.991%</td>
</tr>
<tr>
<td>Approximation 1</td>
<td>4.865</td>
<td>99.992%</td>
</tr>
</tbody>
</table>
Figure 10: \( RPL_c \) (large, bold black circle) with bias errors (dots clustered near origin) and biased noise errors (darker colored dots) shown. The bias offset is also shown (bold black vector and small circle symbol). The approximation \( RPL_{c1} \) (dashed circle) is slightly larger than the exact value, since noise and bias are arranged collinearly.

- Pages 152-153: Fig 29 and Fig 30: vertical axis labels should read, “Protection Level, m”

- Page 195: Strike the following 2 lines:

  % Find the RPL
  RPL_H0 = edcaci3(prob_level,[RPL_H0_bias_east,RPL_H0_bias_north]',Cxy);

Replace them with the following lines:

  % Find the RPL
  % RPL_H0 = edcaci3(prob_level,[RPL_H0_bias_east,RPL_H0_bias_north]',Cxy);
  % Above is a shortened search from method 6 below, but is not as
  % simple a calculation as the one immediately below, which places the
  % bias and noise components in a collinear arrangement:
  RPL_H0 = K_ffmd * HPL_H0_sigma + hypot(RPL_H0_bias_east,RPL_H0_bias_north);
The reader is also referred to the following publications that expand on these topics and present additional results from the data sets used in this dissertation:


The Matlab code in Appendix A is also available online at http://www.mathworks.com/matlabcentral/fileexchange/29639.
ABSTRACT

BRUCKNER, DEAN C., Ph.D., March 2010, Electrical Engineering


Director of Dissertation: Frank van Graas

Four primary contributions are made to the treatment of noise and conspiring bias for dual frequency differential Global Satellite Navigation Systems (GNSSs). These contributions enhance accuracy and protection levels for aircraft precision approach and landing operations and similar applications.

A statistical characterization is presented of Global Positioning System (GPS) user range error as an uncorrelated, normally distributed random variable with non-zero mean over the length of the aircraft precision approach operation. This leads directly to modeling GPS error in the position domain as multivariate normal with non-zero mean.

Based on this model, a vertical composite protection level $VPL_c$ and a horizontal composite protection level $HPL_c$ are each implemented as univariate normal distributions with non-zero means. A method is presented by which exact values—that is, values accurate to a user-defined error tolerance and consistent with statistical assumptions—of $VPL_c$ and $HPL_c$ are obtained, and by which computationally efficient approximations may be evaluated. A statistical quadratic form under the multivariate normal distribution is then used to derive a new class of protection levels based on the probability enclosed within a radius defined in two or more dimensions. A central chi-square representation of this quadratic form is also presented, and is incorporated into a six-step computational
procedure for the two-dimensional composite radial protection level $RPL_c$. This procedure is extended to the composite spherical protection level ($SPL_c$) and the ellipsoidal protection level ($EPL_c$).

Two additional algorithms are presented for dual-frequency differential Global Positioning System (GPS) use. Performance improvements are achieved first through the exchange of pseudorange noise and multipath for reducible biases using a modified Code Noise and Multipath (CNMP) algorithm applied both to reference station and aircraft ranging measurements. In this algorithm, the second frequency is used only to correct the code-minus-carrier (CMC) observable; other ionosphere errors are removed in differential processing. The corrected pseudorange measurements are then combined using a single-frequency carrier phase position domain smoothing (CPDS) algorithm. Composite vertical and horizontal protection levels for the H0 hypothesis are calculated.

The algorithms are implemented in a processing architecture suitable for short-baseline differential operations and termed Dual-Frequency Differential 1 (DFD1). They are then tested on recorded flight test data that includes six aircraft precision approaches. Performance is compared to the LAAS single-frequency architecture. Results over the entire flight data set show an improvement in vertical navigation system accuracy from 0.80 m to 0.32 m (95%), and in horizontal accuracy from 0.41 m to 0.33 m (95%). Mean reductions of the vertical and horizontal protection levels by 60% and 57%, respectively, are observed. Similarly, dramatic improvement is seen in all measures for the six 150-s approaches within this flight, including reduction in 95% error from 0.56 m to 0.26 m vertically and from 0.28 m to 0.14 m horizontally. These all demonstrate the
effectiveness of the composite protection levels and the CNMP and CPDS algorithms within the DFD1 architecture.

This architecture uses C/A code on L1 and carrier phase measurements differenced over time on L1 and L2, but does not resolve integer ambiguities. Nonetheless, it achieves the same level of \(|\mu| + 2\sigma\) navigation system error (NSE) performance in these flight tests as in all published studies of similar class systems known to the author that employ differential kinematic carrier phase architectures between ground and air and require ambiguity resolution.

Approved: _____________________________________________________________

Frank van Graas

Fritz J. and Dolores H. Russ Professor of Electrical Engineering
ACKNOWLEDGMENTS

Thanks so much to the people who have helped, encouraged and mentored me in this remarkable voyage of professional growth and discovery.

First, thank you, Dr. van Graas, for your faithful and skillful guidance throughout these last few years. Your grasp of overarching research strategy and the smallest but most vexing equation is remarkable and inspiring. Thank you for your hospitality too. I hope my future success in some way reflects the considerable investment you have made.

Thank you, Dr. Skidmore, for your trust in me during our project work together, and for helping to provide a bubble of normalcy among the many tasks required of us.

Thank you, Dr. Pelgrum, for developing the real-time TCP/IP interfaces.

Thank you, Dr. Ostermann, Dr. Rankin, Dr. Soloviev, Dr. Skidmore and Dr. van Graas for working out a scholarship and follow-on funding so I could finish the program.

Thank you, Dr. Braasch, Dr. Uijt de Haag, Dr. Vinogradov and Dr. van Graas for teaching me more than I remember, and a little less than I have since learned.

Thank you, Curtis Cohenour, Jay Clark, Kadi Merbough, Tom Brooks, Paul Nilles, Bryan Branham, Jamie Edwards and Aaron Wilson for your timely, practical and professional help with various parts of this research. I hope to return the favor.

Thank you, Sharon Conner and Casey McCartney for your encouragement.

Thank you, colleagues and friends in the Russ College and Avionics Engineering Center, for your friendship, and for making this a wonderful place to be.

And thank you especially, Laurie, for your love and cheerful support. Wow! I couldn’t have done it without you, nor would I have wanted to.
This research was supported, in part, by the Federal Aviation Administration under Aviation Cooperative Research Agreement 98-G-002. Flight test support was provided by the Avionics Engineering Center. This document represents the views of the author and does not necessarily represent the views or policy of the FAA or Ohio University.

I very much appreciate the contributions of my Dissertation Committee, which included Dr. van Graas, Dr. Rankin, Dr. Braasch, Dr. Uijt de Haag, Dr. Glasgow and Col. Coulter, USAF.
To my wonderful wife

Laurie F. Bruckner

and to my parents

Drs. Lee and Lila Bruckner

and to the glory of

God
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>3</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>6</td>
</tr>
<tr>
<td>List of Tables</td>
<td>14</td>
</tr>
<tr>
<td>List of Figures</td>
<td>15</td>
</tr>
<tr>
<td>Chapter 1: Introduction</td>
<td>18</td>
</tr>
<tr>
<td>Protection Levels for High Integrity, High Accuracy GPS</td>
<td>18</td>
</tr>
<tr>
<td>Two Approaches to Defining Protection Levels</td>
<td>20</td>
</tr>
<tr>
<td>Multiple-Frequency Differential GPS for Civil Aviation Use</td>
<td>22</td>
</tr>
<tr>
<td>New Algorithms for Dual-Frequency Differential GPS</td>
<td>22</td>
</tr>
<tr>
<td>Research Questions</td>
<td>24</td>
</tr>
<tr>
<td>A Modified Dual-Frequency Differential GPS Processing Architecture</td>
<td>25</td>
</tr>
<tr>
<td>The LAAS baseline architecture</td>
<td>25</td>
</tr>
<tr>
<td>Modifications to the LAAS architecture</td>
<td>26</td>
</tr>
<tr>
<td>“Turn by Turn” Directions for this Document</td>
<td>31</td>
</tr>
<tr>
<td>Chapter 2: Literature Review</td>
<td>33</td>
</tr>
<tr>
<td>Features of Current and Proposed Differential and Relative GPS Architectures</td>
<td>34</td>
</tr>
<tr>
<td>Fault hypotheses for precision approach and landing</td>
<td>34</td>
</tr>
<tr>
<td>Treatment of residual biases</td>
<td>37</td>
</tr>
<tr>
<td>Vertical and horizontal protection levels</td>
<td>40</td>
</tr>
<tr>
<td>Radial protection levels</td>
<td>44</td>
</tr>
</tbody>
</table>
Conclusions ......................................................................................................................... 164
Resolution of Research Questions .................................................................................... 165
Chapter 10: Future Work .................................................................................................. 167
References ......................................................................................................................... 169
Appendix A: Software for Protection Level Simulation ..................................................... 187
    calculate_protection_levels.m ..................................................................................... 188
    compositePLdemo.m ................................................................................................. 203
LIST OF TABLES

Table 1: Precision approach and landing categories................................................................. 51

Table 2: Summary of achieved accuracy performance of differential and relative GPS architectures in flight tests ..................................................................................... 60

Table 3: Equations for LAAS and composite protection level approximations .................. 96

Table 4: Parameters of Monte-Carlo simulation..................................................................... 97

Table 5: Simulated performance of VPL\(_c\) methods ............................................................ 101

Table 6: Simulated performance of HPL\(_c\) methods .............................................................. 103

Table 7: Simulated performance of RPL\(_c\) methods .............................................................. 104

Table 8: Modified CNMP algorithm parameter values for flight test data......................... 125

Table 9: GSL-C and GSL-D performance .............................................................................. 147

Table 10: Summary of results for entire flight........................................................................ 153

Table 11: Vertical statistics for six approaches (all values in m).......................................... 157

Table 12: Horizontal statistics for six approaches (all values in m)..................................... 158

Table 13: Summary NSE statistics for six approaches (all values in m)............................. 159
LIST OF FIGURES

Figure 1: Illustration of example LAAS airport configuration with three reference antennas and four GPS satellites ............................................................... 26

Figure 2: Overview of LAAS and DFD1 differential GPS processing architectures for reference station ............................................................... 29

Figure 3: Overview of LAAS and DFD1 differential GPS processing architectures for airborne system ............................................................... 30

Figure 4: Mid-value selected data (black bold line) for 100 s of pseudorange corrections from three GPS receivers (SV 2 on Feb. 16, 2009, Athens, Ohio) ......... 36

Figure 5: Probability density function for a biased (i.e., non-zero mean) univariate normal distribution with original and new VPL\textsubscript{c} shown ........................................ 76

Figure 6: Errors enclosed within linear HPL but not enclosed within radial HPL ......... 78

Figure 7: VPL\textsubscript{c1} in x-z plane (dashed black bold), with head-to-tail bias vectors (white solid), and bias and noise projected on vertical axis (dashed red, blue) ........ 89

Figure 8: Illustration of HPL\textsubscript{c1} in horizontal plane, with head-to-tail bias vectors added collinearly to ellipse’s semi-major axis length ........................................ 91

Figure 9: Close-up of worst-case horizontal bias combinations: true (white), Approximation 1 (black), and as magnitude (dashed red) in protection level ...... 92

Figure 10: Close-up of worst-case horizontal bias combinations: true (white), Approximation 1 (black) and Approximation 2 (red dashed) ................. 94

Figure 11: Noise (components and total) versus satellite elevation ........................ 99
Figure 12: Simulation NSE (noise and bias) for east, north and up axes (unsmoothed; 100,000 points) ................................................................. 100

Figure 13: Simulated biased, unsmoothed error in x-z plane, projected onto vertical axis with ±VPL<sub>c</sub> (solid lines) and one approximation to VPL<sub>c</sub>............................. 101

Figure 14: Simulated biased, unsmoothed error in x-y plane, projected onto rotated axis with ±HPL<sub>c</sub> (solid lines) and two approximations to HPL<sub>c</sub>............................... 102

Figure 15: Bias combinations available in the horizontal plane............................... 104

Figure 16: RPL<sub>c</sub> with noise and bias errors shown. The approximation is the same as the exact value, so the second circle is not visible.................................................. 105

Figure 17: Maximum ground multipath error as a function of elevation angle, NovAtel Pinwheel 600 at 3 m height above ground......................................................... 120

Figure 18: CNMP algorithm example: ionosphere-corrected CMC before and after CNMP (top), and total bias bound (bottom). A lock reset occurs at 629 s........... 124

Figure 19: Range difference AO from users at points A and B to satellite at point S and at time t, with parallel lines of sight................................................................. 131

Figure 20: True range difference AC from users at points A and B to satellite at point S, all at time t ................................................................................................. 132

Figure 21: Propagation of position B from time <i>t</i><sub>1</sub> to <i>t</i><sub>2</sub> ................................................................. 134

Figure 22: Ohio University Avionics Engineering Center’s King Air C90SE test aircraft .......................................................... 140

Figure 23: Reference station in Ohio University LAAS shelter............................... 143

Figure 24: King Air ground track for October 29, 2008 flight ............................ 144
Figure 25: King Air precision approach segments for October 29, 2008 flight .......... 145
Figure 26: Number of Satellites, VDOP and HDOP for October 29, 2008 flight ........ 146
Figure 27: Vertical accuracy and 95% error performance for LAAS (with 1 reference receiver) and DFD1 architectures, entire King Air flight test ............................ 149
Figure 28: Horizontal accuracy and 95% error performance for LAAS (with 1 reference receiver) and DFD1 architectures, entire King Air flight test ............................ 150
Figure 29: Vertical protection level performance for modified LAAS and DFD1 (modified CNMP) architectures, entire King Air flight test ............................ 152
Figure 30: Horizontal protection level performance for modified LAAS and DFD1 (modified CNMP) architectures, entire King Air flight test ............................ 153
Figure 31: Vertical error performance for baseline LAAS architecture (top) and the DFD1 architecture (bottom), six 150-s approaches ............................................. 156
Figure 32: Horizontal error performance for baseline LAAS architecture (top) and the DFD1 architecture (bottom), six 150-s approaches ............................................. 157
CHAPTER 1: INTRODUCTION

Protection Levels for High Integrity, High Accuracy GPS

The increasing use of the Global Positioning System (GPS) for high-accuracy and high-integrity navigation applications has focused attention on developing improved methods to account for navigation system error (NSE). This error, defined as the difference between a vehicle’s estimated position and true position at a given time, arises from a number of sources including thermal noise, residual tropospheric and ionospheric delays, satellite clock and orbit errors and multipath (i.e., contamination of direct path signals by reflected signals).

For a system to fulfill its integrity requirement, it must estimate the maximum size of each error source, such that the errors are either contained within safe bounds, or that the probability is sufficiently high that a timely warning is issued when the combination of these errors exceeds safe bounds. Integrity probabilities are typically defined over a specified time period. For example, NSE integrity for an aircraft precision approach operation is commonly defined for a specified approach duration.

Setting a protection level (PL) is a widely-accepted integrity methodology for navigation systems. A PL is commonly defined as the maximum NSE from the true position that can occur in a specified direction (e.g., vertical), “guaranteed” to a sufficiently high level of probability. Integrity specifications in the GPS-based Local Area Augmentation System (LAAS) and Wide Area Augmentation System (WAAS) call for calculating vertical and horizontal or lateral (i.e., cross-track) PLs [1-3]. GPS Receiver Autonomous Integrity Monitoring (RAIM) algorithms likewise compute a
horizontal PL. In these types of applications, the PL is continually compared to an appropriate Alert Limit (AL) throughout the operation. If the PL exceeds the AL, an alarm is annunciated to the user. This typically results in the navigation system being declared unusable for the intended purpose until the PL again drops below the AL.¹

The PL methodology for aviation safety incorporates a major advantage. It is a convenient and pilot-friendly way to visualize an equivalent level of safety, intuitively combining the concepts of accuracy and integrity. For a given level of safety, the PLs together create a high-confidence zone of maximum error surrounding the aircraft’s true position. This zone may be defined in one or more dimensions. Integrity is assured as long as the PL does not exceed the preset AL, which is tailored both to the current phase of the operation and to certain prevailing conditions, such as visibility at the airport. Degraded conditions such as reduced visibility can trigger a tighter AL, thereby limiting operational flexibility. Conversely, system changes may be implemented to increase operational flexibility by reducing the size of PLs. Such changes may involve employing a more sophisticated navigation system, reducing the probability level governing the integrity calculations or revising the method for computing the PLs. Any such improvement is reflected in system metrics such as availability and continuity.

¹ Bernath describes the horizontal protection level as follows: “Instead of a tradeoff between false alarms and missed detections, both probabilities will be traded off against the protection radius, which is the smallest horizontal position error that is guaranteed to be detected with the given probabilities of false alarms and missed detections. Instead of setting an alarm threshold and then calculating the corresponding probabilities, first the probabilities are set and protection radius [is] determined from the probabilities.” [4]
The use of PLs is set to expand. Designers of new high-accuracy relative navigation systems, such as those contemplated for airborne refueling, spacecraft docking, multiple uninhabited aerial systems (UASs) flying in close formation, autonomous ground vehicle convoys, precision farming, highway lane-keeping or even billing systems for toll roads, will increasingly grapple with requirements for high levels of integrity as well. Such applications, along with the desire for increased operational flexibility in existing systems such as LAAS and WAAS, are already stimulating new PL characterizations. Even the traditional focus on the vertical protection level (VPL) over the horizontal protection level (HPL)—appropriate in aircraft precision approach applications, where vertical error dominates—may someday be reversed as other applications mature.

Two Approaches to Defining Protection Levels

Integrity for differential GPS systems such as WAAS and LAAS has traditionally been assured by overbounding both the bias-like errors (i.e., persistent non-zero mean errors) and noise-like errors seen in GPS measurements by using a single closed-form probability distribution. Typically, this has been the normal distribution. A different approach has also been advanced, one that explicitly assigns probability distributions both to noise and to biases. As with the overbounding approach, a PL is produced to guide the aircraft pilot. However, in this approach, the PL is a composite of separate bias and noise distributions.

The presence of biases has continued to complicate the GPS overbounding approach. When sigma values are inflated to cover bias effects, operational flexibility, as
reflected in availability and continuity measures is necessarily reduced. As the next
chapter illustrates, various modifications to the overbounding approach have been
recently proposed. Many of these would require changes to WAAS and LAAS
specifications such as adding dual-frequency receivers and transmitting additional
ground-to-air parameters, including biases. Such changes will likely be accompanied by
new analyses to make best use of new data available in the aircraft. The present effort is
one such analysis. Other differential and relative GNSS applications may benefit as well.

A desire to increase the operational flexibility of high-accuracy, high-integrity
GPS-based navigation systems motivates further development of the composite PL
approach in this document. The literature already reflects various efforts to characterize
the composite vertical protection level ($VPL_c$), such as in van Graas et al. [5] and Rife et
al. [6]. In the present effort, a statistical method for computing $VPL_c$ and a new
horizontal protection level ($HPL_c$) is presented. The existing $VPL_c$ in [5] and the VPL,
HPL and Lateral Protection Level (LPL) for a single faulted LAAS reference receiver in
[1] are shown to be computationally efficient approximations of this statistical method.
The procedure to quantify the degree of excess overbounding of these approximations is
illustrated in simulation. A new class of composite protection levels is developed from a
quadratic statistical form. This class includes the two-dimensional radial protection level
($RPL_c$), as well as the three dimensional spherical protection level ($SPL_c$) and ellipsoidal
protection level ($EPL_c$).
Multiple-Frequency Differential GPS for Civil Aviation Use

As non-military systems, LAAS and WAAS currently augment civilian GPS user measurements via the original L1 civil frequency band, which is centered at 1575.42 MHz. Although the WAAS reference stations employ dual frequency receivers using the L2 band (1227.60 MHz) as described in [7] for ionospheric delay measurement and error reduction, LAAS reference station and airborne receivers currently do not. The advent of civilian-accessible GPS ranging signals at the L5 (1176.45 MHz) frequency as described in [8] opens up new possibilities for improved differential performance for civil aviation users. Ionospheric delay is a potentially large differential GPS error. Through the use of multiple-frequency measurements, ionospheric delays at the reference station and aircraft may be determined and removed from measurements prior to differential processing, or may be used in differential processing in other ways. Thus, the use of dual or triple-frequency measurements should be considered as a possible major capability upgrade for differential GPS systems.

Through the GPS Precise Positioning Service (PPS) that provides precise code measurements on both L1 and L2 frequencies to approved users, military differential systems are inherently dual-frequency. The processing architecture described in this document may provide additional benefit for these systems as well, but such improvements are beyond the scope of this study.

New Algorithms for Dual-Frequency Differential GPS

Besides the recasting of protection levels as composites of noise and bias, another outgrowth of the separate statistical treatment of these two error types has been the
development of techniques to pit noise and bias errors against each other, in an effort to reduce both. One such technique is the Code Noise and Multipath (CNMP) algorithm currently used in the WAAS reference stations [9-10]. This algorithm creates a dual frequency code-minus-carrier (CMC) observable to characterize and remove the pseudorange noise, pseudorange multipath and time variations in ionospheric delay that occur after, and are referenced to, the value at algorithm initialization. This processing is performed by both reference and airborne receivers. Subtracting the CMC observable from the pseudorange to remove these error sources introduces a pseudorange bias. This bias is unknown, but may be estimated by time averaging and bounded by analysis and observation of multipath signatures in the CMC observable. The merit of this approach is demonstrated by showing that the maximum bound on introduced pseudorange bias, which decreases with time, is smaller than removed errors. The goal is to maximize “profit” (i.e., net reduction in error and protection levels) in this exchange.

Similarly, a carrier phase position domain smoothing (CPDS) algorithm summarized in [11] harnesses the stability of continuously tracked carrier phase measurements. Instead of using these measurements to smooth the pseudoranges as in conventional techniques, however, the algorithm exploits them to propagate the user’s position with centimeter level accuracy. The thorn of code-carrier divergence that occurs in range domain processing is removed, and the long-term error in propagated position solution is kept close to zero by continual updates from pseudorange measurements. In effect, position domain smoothing creates a fixed “rail in the sky” to define the aircraft approach path, effectively smoothing the jumps in position domain biases that occur
when satellites transition into or out of the visible constellation. Carrier phase measurements from new satellites entering the visible constellation can be used immediately to propagate the user’s differentially corrected position.

**Research Questions**

The analysis of [6] demonstrated that even with a new LAAS ground-to-air data message structure, code phase measurement biases as small as 8 cm will still have an impact on system availability for LAAS Category III (i.e., the most stringent precision approach) for the worst-case airports in the continental United States. Similarly, biases of 80 cm or larger degrade availability at the typical airport in the continental U.S. Besides this, the necessity of extremely low position errors in future operations such as autolanding of aircraft provides a motivation to reduce these errors to the lowest level possible.

Against the background of these requirements for performance improvements, the following three research questions are formulated:

1. Can protection levels be computed exactly—to a specified error tolerance of arbitrary size—for a set of non-zero mean, multivariate normal random variables that represent GPS position domain error?

2. Can a modified differential GPS processing architecture significantly reduce position error and protection levels in flight tests compared to a conventional LAAS architecture in similar conditions?

3. Can a modified differential GPS processing architecture that does not employ carrier phase integer ambiguity resolution attain same, or nearly the same, level of
accuracy performance as a kinematic carrier phase processing architecture that does?

The remainder of this document explores answers to these questions. The primary modifications to the LAAS differential GPS processing architecture, which serves as a baseline, are made by treating noise and bias separately instead of overbounding them together as noise-like errors.

A Modified Dual-Frequency Differential GPS Processing Architecture

A modified differential GPS processing architecture is proposed to demonstrate the advantage of treating noise and bias separately. Since the term “modified” is used in reference to the LAAS baseline architecture, a brief explanation of that system is appropriate first.

The LAAS baseline architecture

The LAAS reference station architecture, as defined in [1], is based on up to four GPS antenna-receiver pairs situated in a low-multipath environment and with a one-way datalink from the reference station to the aircraft. Ideally, this installation is co-located with the primary airport it serves to maximize differential GPS performance, but this is not a requirement in general. The arrangement appears in Figure 1 below for three reference antennas and four satellites. The VHF data broadcast uplink is also shown. The pseudorange corrections from each antenna-receiver pair for each satellite are resolved to a single antenna location and averaged, and then typically smoothed using an alpha filter for a time interval of up to 100 s as detailed in chapter 6. The results are transmitted on the data broadcast for use by the airborne system, which measures and smoothes its own
satellite measurements according to the same procedure. The differentially corrected, single-frequency GPS position solutions and protection levels are calculated and used in the precision approach operation. As the distance between the aircraft and the reference station decreases, so do errors that arise due to spatial decorrelation between the two, such as residual ionospheric and tropospheric delays. The result is a high-accuracy, high-integrity navigation solution suitable for safety-critical, civil aviation applications.

**Figure 1: Illustration of example LAAS airport configuration with three reference antennas and four GPS satellites**

*Modifications to the LAAS architecture*

The physical components of the modified differential GPS system are the same as those illustrated for LAAS in Figure 1, except that dual-frequency antennas and receivers are used. The processing architecture, however, differs in several ways. The data broadcast from reference station to aircraft includes not only the differential corrections,
but also pseudorange measurements, accumulated Doppler measurements, noise sigmas and bias bounds, all of which enable enhanced aircraft processing. The modified differential GPS processing architecture also incorporates a modified version of the CNMP algorithm, namely one that calculates bias bounds based on real-time observations rather than preset values, and is applied in the aircraft as well as the reference station. Dual-frequency measurements are used only to remove time variations in ionospheric delay—that is, delay that has occurred since CNMP algorithm initialization—from the CMC observable. They are not used to correct code phase and carrier phase measurements for ionospheric delays. Only carrier phase data is used from L2; code phase data from L2 is discarded. Absolute ionospheric delay is removed only as a result of differential processing, and residual ionospheric error tends to zero only as the aircraft approaches the reference station and “sees” the same ionosphere. Both the CPDS and composite PL algorithms are included in the modified processing architecture, which is referred to simply as Dual-Frequency Differential 1, or DFD1 for short.

The DFD1 architecture exhibits three additional distinctive features. First, the least-squares position solution is weighted by the inverse of the maximum bias bound for each satellite. This weighting technique is selected because post-CNMP noise sigma values are much smaller than corresponding maximum bias bounds. At least one prior study has shown that weighting by the inverse of noise variance is ineffective where significant pseudorange biases exist [12]. However, marked improvement is seen with bias weighting. To ensure the weighting matrix has full rank, bias bounds are restricted to the range $0.01 < B < 1000$ m, where $B$ is the bias value.
A second feature is that range domain observables are broadcast from the reference station without the application of conventional range domain smoothing methods. This is necessary for the position domain smoothing technique employed in the aircraft, as it avoids introducing new biases that might occur if a severe ionospheric spatial gradient exists between the reference station and aircraft during smoothing.

Third and finally, the H1 hypothesis reserved for the case of one malfunctioning GPS reference receiver is removed by operating three antenna-receiver pairs in parallel and implementing a mid-value select algorithm. This guarantees a fault-free signal available for navigation processing, even if a fault occurs in one reference receiver. As a consequence, PLs are calculated in this study only for the null hypothesis, H0. Using only the H0 hypothesis in analyses of this type has precedent, as documented in [13-14]. However, this approach is chosen primarily because reductions in ranging source error—one of the few variables that the algorithm designer can influence—are so closely linked to H0 protection level calculations, which in turn so greatly influence system availability and continuity. It is also assumed a certain level of bias can and does occur in normal GPS operation, and its mere presence does not necessarily lead to the rejection of the H0 hypothesis. The flight test data used here was obtained for a single reference antenna-receiver pair, and so the full realization of this architecture is offered for future research.

The similarities and differences between the LAAS and DFD1 processing architectures are illustrated in Figure 2 and Figure 3 below:
Figure 2: Overview of LAAS and DFD1 differential GPS processing architectures for reference station
Figure 3: Overview of LAAS and DFD1 differential GPS processing architectures for airborne system
“Turn by Turn” Directions for this Document

The remainder of this document is devoted to explaining how explicit treatment of noise and bias in the design of the major components of the DFD1 processing architecture yields significant overall performance improvements. Chapter 2 contains a survey of the literature on key features of selected differential and relative GPS systems, and a summary of related innovations offered by the DFD1 processing architecture. In chapter 3, a statistical characterization of VPL_c and HPL_c is developed, and an algorithm by which these PLs may be computed to a user-defined probability error tolerance is described. The radial protection levels RPL_c, SPL_c and EPL_c are derived. In chapter 4, it is shown that existing formulations of composite protection levels are computationally efficient approximations to this new method. Simulation results are presented to illustrate how the degree of excess overbounding in these approximations may be quantified.

Chapter 5 contains the definition of a modified CNMP algorithm. A worst-case error bound on bias uncertainty is developed, and the minimum initialization time of the algorithm is derived from the lowest multipath fading frequency. The CPDS algorithm is derived in chapter 6, along with a summary of the LAAS 100-s ranging smoothing algorithm that serves as a performance baseline.

The flight test setup is described in chapter 7. The cumulative improvement of all the explicit noise and bias techniques defined in earlier chapters is clearly demonstrated in the flight test results of chapter 8. Conclusions are stated in chapter 9, and Future Work is recommended in chapter 10. Finally, Appendix A contains complete software code
listings for the protection level calculations and Monte-Carlo simulations. The software was written in Matlab®.  

The material in chapter 3, along with selected portions of chapter 1, 2 and 9, also appears in [15]. Similarly, the material in chapter 4 appears in [16]; the material in chapters 5 through 8, along with a portion of chapter 9, also appears in [17]; and the material in chapter 6 appears in [18].

---

2 Matlab® is a trademark of The MathWorks, Inc., Natick, Massachusetts. The ® symbol is omitted hereafter for readability.
CHAPTER 2: LITERATURE REVIEW

An abundance exists of differential and relative GPS architectures for civil private, civil government and military applications. This chapter primarily describes published descriptions of differential GPS architectures promulgated by United States civil government agencies for precision approach and landing operations by commercial aircraft, and suggested modifications to these architectures. A few references reach beyond this focus where appropriate. Alongside these descriptions of current architectures, the innovative features of the proposed DFD1 architecture are highlighted. The areas examined include fault hypotheses, treatment of residual biases, vertical and horizontal protection levels, radial protection levels, the modified CNMP algorithm, carrier phase smoothing techniques, civil and military dual-frequency differential GPS implementations, relative GPS architectures and a few other selected applications. The significance of the DFD1 architecture is also discussed.

A few current and planned differential GNSS architectures exist outside the United States for safety of life applications such as civil aeronautics, but none are as fully developed as the systems examined in this chapter. For this reason, non-U.S. GNSS architectures are not included in the limited space allocated in this chapter to literature review. However, the techniques described in this document are generally applicable to high-accuracy, high-integrity GNSS applications worldwide.
Features of Current and Proposed Differential and Relative GPS Architectures

Fault hypotheses for precision approach and landing

Integrity specifications for systems such as LAAS and WAAS include hypothesis tests that are continually performed throughout the precision approach and landing operation. In the example of LAAS, which is a Ground-Based Augmentation System (GBAS) to GPS, integrity risks are divided into three hypotheses, as specified in [1]. The H0 hypothesis, or null hypothesis, assumes that the ground-based data broadcast signal is available at the aircraft in its normal operating condition. The H1 hypothesis assumes that a fault has occurred in exactly one ground-based GPS reference receiver, that this fault has produced misleading information (i.e., a bias in the position-domain error distribution), and that the fault has not been detected. A PL is calculated for each of these two hypotheses. These two PLs are then compared, and the largest one is selected for use. The H2 hypothesis covers all other undetected faults that have similarly produced misleading information, but does not enter into the PL calculation.

As in Rife et al. [13] and Shively [14], this effort examines PLs only under an H0 hypothesis. This is appropriate for the analysis of a generic PL formulation not closely tied to the configuration of any particular system such as LAAS or WAAS. It is also appropriate in order to permit a focus on improving system performance through error reduction. As noted in [14], availability improvements come from four sources: increasing alert limits, increasing the number of ranging sources, decreasing the probability of ranging source failure, and reducing H0 protection levels. Only the final factor may be influenced by the algorithm designer. The way in which H0 protection
levels are computed requires ranging source errors to be reduced as a condition for improvement. Thus, the simplest and best criterion for comparison is the H0 hypothesis, particularly for an initial analysis.

With that said, the DFD1 architecture is expected to excel in the single receiver fault scenario as well. With a mid-value select algorithm using three antenna-receiver pairs, a fault-free output is assured if one reference receiver malfunctions. Faulty data is, by definition, always abnormally higher or lower than the true value. Thus, it cannot be the middle value along with values from two fault-free receivers, and is not selected for transmission to the aircraft. Thus, the hypothesis set is streamlined by the removal of the H1 hypothesis altogether.

The mid-value select algorithm is illustrated in Figure 4 by a 100-s segment of data from three NovAtel OEM-4 receivers connected to a single NovAtel PinWheel 600 antenna at the Ohio University LAAS reference station. In this data segment from February 16, 2009, a mid-value select algorithm has been applied to the unsmoothed pseudorange corrections from GPS satellite 2. The pseudorange, which is a range measurement for which the user receiver’s clock error has not yet been estimated and removed, is the primary observable in single-frequency LAAS.

For $m = 3$ reference receivers, the mean-value algorithm used in conventional LAAS will reduce the noise variance by $1/m = 0.33$, under the assumption that the noise is uncorrelated between the reference receivers. In the mid-value select algorithm, an approximation for the reduction in variance from an $m$-point median filter with Gaussian noise input is given by [15] as $(\pi/2)/(m + \pi/2 - 1)$. For $m = 3$, this is a factor of 0.44.
Thus, the variance reduction effect of the mid-value select algorithm is not as great as that of simple averaging. However, it is evident that the measurements from the reference station will not be corrupted by a malfunction from a single receiver.

Figure 4: Mid-value selected data (black bold line) for 100 s of pseudorange corrections from three GPS receivers (SV 2 on Feb. 16, 2009, Athens, Ohio)

The H0 hypothesis governing the present analysis is simply normal GPS system operation, with one additional proviso. It is assumed that a certain amount of range-domain bias occurs in user measurements in normal GPS operation. This point has been illustrated in recent years in various studies of residual group delay bias in GPS antennas
and antenna arrays [19]. The mere presence of such bias, therefore, does not necessarily lead to the rejection of this H0 hypothesis and the subsequent declaration of a system fault. 

No bias fault detector is explicitly implemented in the proposed architecture, as in H1 processing for LAAS. If the computed bias bound is consistent across the three receivers but is excessively large, the PL will exceed the AL. This will automatically result in the system being declared unusable until the PL is again within the AL. To detect multiple simultaneous ground receiver failures that do not exceed the alert limit, a monitor should be implemented to compare the three outputs periodically for consistency.

**Treatment of residual biases**

Almost since the inception of GPS, it has been recognized that the system's user range domain errors generally fall into the categories of persistent bias-like errors and random noise-like errors, as in Greenspan et al. [20] and Pervan et al. [21]. The range domain is the portion of GPS signal processing in which distance measurements between a user and multiple satellites are estimated. Range domain errors propagate via linearized transformations into the position domain, where the user's geographic position, velocity and time are estimated. Noise-like range domain errors seen by the GPS user have generally been modeled as zero mean, uncorrelated, normally distributed random variables, both for the sake of simplicity and to avoid imposing a more complex model whose modeling error might compound the characterization problem [22]. Small bias-like errors have been treated in the same way, with sigma inflation applied as needed.
With the end of GPS Selective Availability’s performance degradations in May 2000, and the growing need to find conclusive error bounds to meet demanding LAAS and WAAS accuracy and integrity requirements, much attention has been applied to characterizing bias errors that still remain after error removal techniques are applied. Along the way, two general schools of thought have emerged on how best to do this.

The most prominent approach to characterizing residual range-domain biases has been to overbound them using a single probability distribution expressed in closed form. Typically, a zero mean normal distribution has been selected, with sigma values inflated as required to account for the biases not formally included in the model. The intent of this approach has been to ensure that a conclusive probability bound can be applied to very rare, and possibly catastrophic, error events. With acceptable probabilities of hazardously misleading information (HMI) in LAAS Category III aircraft landing approaches on the order of $1 \times 10^{-9}$ per 15-second period, the weaknesses of simulation and empirical data analysis in characterizing such rare events—as well as the attractiveness of closed-form solutions—have been obvious. Contributors to this approach included Shively, DeCleene, Pervan, Pullen, Enge, Rife, Walter and others [23-29].

An alternate approach has also emerged, one that calls for separation of bias-like and noise-like errors throughout the GPS data processing chain. This approach has included an underlying assumption that maximum residual biases seen by the user in normal GPS operation can be bounded conclusively through analysis and, ultimately, through safety assessment techniques of the type detailed in [30]. This composite PL approach led to the development of the VPL-C (here termed VPLc) by van Graas,
Skidmore and a few others [5, 12, 31-32]. Efficient algorithms for bounding position
domain error with simple metrics, assuming that a range-domain bound already existed,
were developed by Lee, Loh and Fernow [33-34].

Recently, the overbound approach and the composite PL approach have begun to converge. Concerns about the limited system availability associated with the overbound approach and recognition that new GPS antenna array systems often introduce relatively large group delay biases in normal operation have been accompanied by calls for various modifications to the approach. These have included the following:

- Treating the ionosphere as a residual VPL bias in WAAS-like Space-Based
  Augmentation Systems (SBASs) for mid-latitude regions of the world [35]
- Including an explicit bias term in the real-time pseudorange corrections computed
  by a WAAS VPL local area monitor [36-38]
- Broadcasting LAAS sigma inflation factors to cause the airborne processor to
  reject sub-geometries deemed unsuitable due to ionospheric anomalies [39]
- Calculating a pseudo-bias as part of error bounding by paired distributions for
  LAAS and WAAS, followed by additional overbounding [40]
- Computing LAAS fault-mode error bounds in the aircraft so that actual user
  parameters can replace worst-case assumptions made on the ground [41]
- Broadcasting noise and bias parameters on the new GPS L5 signal to reduce the
  largest WAAS protection levels by an average of 20 percent [42]

In addition to these, the GNSS Evolutionary Architecture Study (GEAS) Panel, tasked with analyzing expected GPS capabilities in the 2025-2030 time frame, recently created
an explicit bias term in its GNSS error model. This is implemented using both a nominal bias value for accuracy and a conspiring bias formulation, which uses maximum bias bound magnitude and appropriate positive or negative signs, for protection levels [43].

The composite PL approach, in turn, has served as a performance baseline for studying the availability of overbounding methods that do not require transmission of bias information [6]. Since it appears that existing augmentation systems may in the end require transmission of a bias parameter, to be implemented via a specification change, the composite PL approach deserves another look. This document is a step in that direction. The relative simplicity of the composite PL statistical characterization, counterbalanced by the rigors of a safety assessment effort that may be needed to ensure a high-probability maximum bias overbound, should be carefully considered.

**Vertical and horizontal protection levels**

The LAAS and WAAS specifications contain the current VPL formulation for the H0 hypothesis [1-3]. This VPL, termed the nominal VPL in [6], is defined as

\[
VPL_{H0} = K_{\text{ffmd}} \sqrt{\sum_{i=1}^{M} S_i^2 \sigma_i^2}
\]

where

\[
K_{\text{ffmd}} = \text{the fault-free missed detection multiplier, directly obtained from the required probability level (unitless)}
\]

\[
M = \text{the number of user-to-satellite range measurements in the solution}
\]
\[ S = \text{the } (4xM) \text{ matrix of partial derivatives } s \text{ that transform the range domain to the position domain in a locally level frame (e.g., east-north-up, plus time) in the least-squares sense} \]

\[ \sigma_i = \text{the } i\text{-th satellite sigma for range domain error (m)} \]

The term \( \sigma_i \) is computed for uncorrelated Gaussian error as follows, and may be inflated to ensure integrity in the presence of biases with non-Gaussian behavior. These error terms are appropriate for differential or relative navigation systems, but may vary somewhat depending on the application (e.g., LAAS vs. WAAS):

\[
\begin{align*}
\sigma_i^2 &= \left( \sigma_{pr, gnd, i}^2 + \sigma_{air, i}^2 + \sigma_{iono, i}^2 + \sigma_{tropo, i}^2 \right) \\
\sigma_{pr, gnd} &= \text{the standard deviation for reference station errors, including noise and multipath (m)} \\
\sigma_{air} &= \text{the standard deviation for airborne errors, including noise and multipath (m)} \\
\sigma_{iono} &= \text{the residual ionospheric delay (due to spatial decorrelation) uncertainty (m)} \\
\sigma_{tropo} &= \text{the residual tropospheric delay (due to spatial correlation) uncertainty (m)}
\end{align*}
\]

The fault-free horizontal protection level defined for aircraft using the LAAS differentially-corrected positioning service, which is a more generic differential formulation compared to the LAAS approach service, is defined below. This formulation is essentially the same as that of the WAAS HPL in [3]:
\[ HPL_{110} = K_{fimd} d_{major} \] (3)

where

\[
d_{\text{major}} = \sqrt{\frac{d_x^2 + d_y^2}{2} + \left( \frac{d_x^2 - d_y^2}{2} \right)^2 + d_{xy}^2}
\]
is the 1-sigma uncertainty of error in the direction of the error ellipse’s semi-major axis, and implements the worst-case horizontal error when multiplied by \( K_{\text{fimd}} \) (m)

\[
d_x^2 = \sum_{i=1}^{N} s_{1,i}^2 \sigma_i^2
\]
is the error variance in the \( x \) (east) direction (m\(^2\))

\[
d_y^2 = \sum_{i=1}^{N} s_{2,i}^2 \sigma_i^2
\]
is the error variance in the \( y \) (north) direction (m\(^2\))

\[
d_{xy} = \sum_{i=1}^{N} s_{1,i} s_{2,i} \sigma_i^2
\]
is the error covariance between \( x \) and \( y \) (m\(^2\))

\( s_{1,i}, s_{2,i} \) = the projections of the east and north components for \( i \)-th ranging source, respectively, from matrix \( S \) as defined above (unitless)

A locally level coordinate frame is used, where \( x \) is east, \( y \) is north and \( z \) is up. These LAAS PLs serve as the baseline to measure performance improvements offered by the VPL\(_c\) and HPL\(_c\).

The sigma values for VPL and HPL defined above are considered an overbound of both noise and bias errors. No explicit bias term is present, but sigma inflation is used to accomplish the overbound. In contrast, the composite VPL of [5], which is termed \( VPL_{bias} \) in [6], defines a sigma that overbounds Gaussian noise only and includes no inflation factor for added bias. The composite VPL accounts for bias errors instead by
using an explicit bias term in the equation associated with the H0 hypothesis. This results in the worst case bias combination, also called a group of “conspiring” biases, being projected onto the vertical axis:

\[
VPL_c = K_{\text{find}} \sqrt{\sum_{i=1}^{M} s_{3,i}^2 \sigma_i^2} + \sum_{i=1}^{M} |s_{3,i} B_i |
\]  

(4)

where

\[
B_i = \text{the } i\text{-th satellite measurement’s range domain bias present in normal (i.e., fault-free) operation (m)}
\]

The noise term of the VPL\(_c\) is assumed to be normally distributed, as is the identical term in the VPL defined previously. The bias term is assigned a truncated distribution, such as the uniform distribution or even a distribution idealized as two delta functions, one each at the upper and lower bias bound ±\(B_{\text{max}}\). If the bias error is sinusoidal, for example, most of the error will be concentrated near these limits, making this distribution a reasonable choice. Addition of these two error sources in a GPS pseudorange measurement is modeled by the convolution of these two distributions. In the case of the double delta distribution, a pair of normal distributions, shifted left and right by \(B_{\text{max}}\), results. This bears a strong resemblance to the paired overbounds defined by Rife et al. in [40]. Empirical estimates of this \(B_{\text{max}}\) overbound value for Category III LAAS are currently in the range of 10 – 20 cm [5-6]. The analysis in [6] examined availability impacts of induced bias in 2-cm steps from 0 to 80 cm, and found that availability benefits of a VPL\(_c\) implementation are most significant for residual (i.e., post-
differential correction) biases larger than 20 cm. This analysis did not address trading the bias for a reduction in the noise.

In chapter 3, a statistical framework is provided within which the composite PLs may be implemented. A simple method, which is based on the underlying univariate normal distributions of VPL\textsubscript{c} and HPL\textsubscript{c}, is presented. By this method, exact values—that is, values accurate to a user-defined error tolerance and consistent with statistical assumptions—of VPL\textsubscript{c} and HPL\textsubscript{c} may be obtained, and approximations to these exact values may be evaluated.

**Radial protection levels**

The HPL has also been defined in the literature in the context of Receiver Autonomous Integrity Monitoring (RAIM) algorithms. A RAIM capability allows an aircraft GPS receiver to detect satellite faults apart from notifications from any integrity monitor present in GPS itself or within a GPS augmentation. In implementations such as those in [44-46], a test statistic is formed, typically involving a sum of squares of error residuals, and the statistic is projected onto the horizontal plane. A quadratic form, in these cases a root-sum-square error distance, is used to form an error radius. The radius is then compared to an alert threshold.

While the uses of a quadratic form and chi-square probability distributions are similar in these conceptual radial protection levels and the radial protection levels defined in this document, several key differences exist. First, the RAIM-based HPLs were envisioned for en-route and non-precision approach operational phases with a 24-satellite constellation while selective availability was still active. Consequently, the HPLs were on
the order of 100 to 300 m instead of the 10-m level needed in precision approach.

Second, these RAIM formulations are constructed in a test statistic domain, not directly in the position domain. Third, the presence of a detectable bias in a RAIM-based HPL is evidence of a satellite fault, not a recognized statistical parameter. Fourth and most importantly, the use of standalone aircraft GPS measurements instead of an integrity-assured differential solution between a reference station and a compatible airborne system places these in separate performance classes. For these reasons, the radial RAIM-based HPL formulations are noted as being similar in some respects to the radial protection levels in the current discussion, but not directly relevant.

The calculation of radial error for a zero mean, bivariate normal distribution is known in the literature for navigation problems [47]. With respect to GPS, a recent stochastic modeling study of kinematic differential GPS error contains analytical integration of the bivariate normal distribution within a 1-sigma error ellipse. This is used to calibrate Monte-Carlo simulation parameters [48]. The error estimator is a radial measurement (i.e., a quadratic form) of absolute position error of the reference station in earth-centered, earth-fixed (ECEF) coordinates \( x, y \) and \( z \). Zero mean Gaussian measurement noise in the range domain is assumed. The variance and an rms combination of mean error of the radial estimate are calculated for different simulated time averaging lengths and range domain measurement noise variance for 100 Monte-Carlo iterations. However, the probability distribution of the position domain radial measure is not specified. Furthermore, because the means of the 100 iterations in each Monte-Carlo run are summed only in an rms sense, the characteristics of the first moment
of the radial estimator are not fully developed (i.e., is the simulation error zero mean in the position domain?). Additionally, the focus of the error measure—whether it highlights absolute or relative navigation performance—is unclear. A reformulation of the error measure, possibly as the integration of radial error under a trivariate normal distribution, would strengthen the theoretical performance estimate of the kinematic differential GPS algorithm proposed in [48].

The only other two similar navigational uses of a quadratic form and chi-square computational algorithm that are known to the author are described in [49-50]. These are, respectively, a 1987 ballistics study and a 2008 aircraft collision-avoidance analysis that approximates projected penetration of a safety sphere with the computationally easier crossing of a horizontal disc boundary. Neither of these studies develops an error protection level of the type defined in this document.

In recent work in relative navigation, such as the air-to-air UAS refueling operation described in [51], position-domain radial error measures are increasingly used, although without protection levels. In this reference, a mean radial spherical error of 0.33 cm is cited. In the emerging context of high-integrity, high-accuracy relative navigation applications, implementation of the new class of radial protection levels derived in this document directly in the position domain is a natural next step.

**Code Noise and Multipath algorithm**

The CNMP algorithm was developed and implemented for use in the WAAS reference stations [9-10]. The algorithm estimates the GPS pseudorange noise and multipath, which are major error sources at the WAAS reference stations, using dual-
frequency CMC observables. The CNMP algorithm has been instrumental in preserving WAAS continuity and availability by reducing protection levels and false alarms, both of which are strongly influenced by the pseudorange noise and multipath. The continuing value of the CNMP algorithm in WAAS was recently confirmed, along with a description of its use in a triple processing thread architecture, in [52].

The roots of the CNMP algorithm reach back to CMC processing techniques being developed years before they were implemented in WAAS. For example, a Code/Carrier Integrity Monitor (CCIM) technique described in [53-54] used a time-averaged CMC observable to estimate the constant CMC bias in real time, similar to the later WAAS implementation of [9-10]. The CCIM was initialized over an $N$-sample period of zero mean multipath, and a sliding, fixed-length averaging window provided subsequent updates.

Since its WAAS implementation, the CNMP algorithm has been studied, as in [55], and implemented in simulation [56]. However, unique modifications to this algorithm have not been presented in the literature. The CNMP algorithm presented in chapter 5 is a modification to the WAAS CNMP algorithm in that bounds on the maximum bias are calculated not as a function of the passage of time according to a fixed formula, but instead are based on maximum and minimum values of the smoothed CMC observable obtained over a specified time interval. In addition, the initial maximum bound, which is set to 10 m for the WAAS reference station, is recalculated for the flight test configuration specified in chapter 7. Moreover, the procedure used for deriving these bounds is clearly specified, so that bounds for a different configuration may be readily
obtained. Thus, the utility of the CNMP algorithm for non-WAAS implementations is expanded.

Although the CNMP processing is corrected for ionospheric spatial decorrelation based on the dual-frequency observable, the ionospheric delay is not actually removed from either code phase or carrier phase measurements in the DFD1 processing architecture. Thus, this architecture is currently suited for short-baseline distances up to approximately 100 m between reference station and runway threshold. Further modifications allowing larger baseline distances are left for further research.

**Carrier phase position domain smoothing techniques**

Using carrier phase measurements to smooth GPS pseudoranges has been a well-known technique first reported in 1982 by Hatch [57]. However, the use of carrier phase data to smooth position or velocity estimates derived from GPS pseudoranges has been comparatively rare. In perhaps the first flight test campaign using this technique, a delta-carrier phase update was applied to unsmoothed pseudorange corrections in differential GPS autoland tests in late 1994 and early 1995 [11]. In the 1995 tests, which involved fifty precision approaches with a United Parcel Service Boeing 757 aircraft, a $|\mu| + 2\sigma$ error performance of 1.1 m vertical and 1.4 m lateral was achieved. The details of this smoothing algorithm were not provided in [11], but details of an updated version are now provided in chapter 6.

At least two additional position domain smoothing algorithms are described in the literature. In the first one, a single GPS receiver with dual frequency capability provides ionosphere-free measurements consisting of undifferenced pseudorange and time-
differenced carrier phase for precise point positioning in post-processing [58]. A CMC observable is used to estimate variance-covariance matrix elements for code noise and multipath, and a variance-covariance matrix for carrier phase terms is also created. A kinematic, sequential least-squares filter, which is a special case of a Kalman filter, uses as innovations the differences between predicted and measured pseudorange as well as the differences between predicted and measured carrier phase updates. This formulation, which uses precise orbit products supplied in post-processing, achieves sub-meter-level positioning accuracy in post-processed static, aircraft and low-earth orbit satellite data sets.

The second formulation, likewise a Kalman filter, is described in [59] and [60]. The state vector consists of the error states of current position, current velocity and previous position. The innovations are the single differences between measured and theoretical pseudorange, and between measured and theoretical Doppler, for two different satellites. A double difference is then formed within the Kalman filter by differencing these across the current and previous GPS epochs. This algorithm is shown in [59] to improve both availability and accuracy for land vehicle use, achieving non-differential mode accuracies of 1-2 m with good geometries and 5-10 m in urban canyons. In [60], the position-domain smoothing is accomplished at both reference station and aircraft where measurements can be made divergence-free or ionosphere-free (terms fully defined in that reference), and only in the aircraft otherwise. Factors listed for consideration when selecting range domain or position domain smoothing included the following:
Filter memory when transiting an ionospheric front (memory hampers position domain smoothing due to “left over” errors after crossing through the front)

The possibility of frequent satellite blockages (blockages favor position domain smoothing as long as four satellites are always visible)

Possible step changes in the position solution due to smoothing if the source of differential corrections changes (i.e., when transitioning to a different reference station)

The position domain smoothing algorithm presented in chapter 6 of this document is not a Kalman filter implementation. Instead, an instantaneous integrated velocity estimate of very high quality is created for the aircraft, and then applied only in the aircraft. Measurements only from the previous and current GPS epochs are needed. Using this estimate, an alpha filter smoothes the differential position that is derived from pseudorange measurements at the reference station and aircraft. Ionosphere error is currently removed only in differential processing, making this proposed architecture currently suited for short-baseline distances of up to about 100 m between runway threshold and reference station. At no time are the pseudorange measurements themselves smoothed using carrier phase information. With no filter memory that can retain inaccurate ionospheric delay information, pseudorange noise that is reduced to a few cm by the CNMP algorithm, and pseudoranges that are weighted by 1 / (bias bound) in the least-squares position solution, the shortcomings of position domain smoothing as described above are resolved. Furthermore, carrier phase measurements from a newly acquired satellite may be used immediately in the position domain smoothing algorithm
while incurring jumps of only a few cm or less with a typical GPS constellation. As long as continuous carrier phase tracking is maintained for at least four satellites, or an inertial measurement unit provides data to coast through momentary blockages, the algorithm continues without interruption.

In attempting to create optimal architectures for future GPS in the 2025-2030 time frame, the GEAS Panel has previously recommended two relative RAIM (RRAIM) architectures. One uses a range domain formulation, and the other employs an accumulated carrier phase based projection of the user’s position as its observable in the position domain [43, 61]. An explicit goal of the recommended second architecture is to reduce the size of PLs.

**LAAS implementations**

The LAAS program supports precision approaches of various categories. These categories, established by the International Civil Aviation Organization (ICAO) in [62] and repeated in [1], are reproduced in Table 1 below:

<table>
<thead>
<tr>
<th>Category</th>
<th>Decision Height (DH)</th>
<th>Runway Visual Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>≥ 200 ft (60 m)</td>
<td>≥ 1,800 ft (550 m) or visibility ≥ 0.5 statute mile (800 m)</td>
</tr>
<tr>
<td>II</td>
<td>100-200 ft (30-60 m)</td>
<td>≥ 1,200 ft (350 m)</td>
</tr>
<tr>
<td>III</td>
<td>&lt; 100 ft (30 m) or no DH</td>
<td>&lt; 1,200 ft (350 m)</td>
</tr>
<tr>
<td>IIIa</td>
<td>&lt; 100 ft (30 m) or no DH</td>
<td>≥ 700 ft (200 m)</td>
</tr>
<tr>
<td>IIIb</td>
<td>&lt; 50 ft (15 m) or no DH</td>
<td>150-700 ft (50-200 m)</td>
</tr>
<tr>
<td>IIIc</td>
<td>No DH</td>
<td>&lt; 150 ft (50 m)</td>
</tr>
</tbody>
</table>
In 1994 and 1995, the FAA sponsored an extended flight test campaign to demonstrate Category IIIb precision approach capability as part of the LAAS program [63]. Two commercial firms and two universities each fielded demonstration systems, completed at least 100 precision approaches and reported NSE values at an altitude of 50 ft height above the runway threshold (HAT). The accuracy values listed in this section for these more than 400 precision approaches are $|\mu| + 2\sigma$ NSE at 50 ft HAT.

Wilcox Electric implemented a C/A code-based differential solution with pseudoranges and pseudorange rates and achieved accuracy performance of 1.30 m vertical and 0.90 m lateral (unfiltered) using a Boeing 727-200 aircraft [64]. Ohio University also implemented a code phase differential solution, but with a carrier phase differential velocity, achieving vertical accuracy of 1.1 m vertical and 1.4 m lateral on the second set of 50 approaches with a Boeing 757 aircraft [11].

E-Systems demonstrated a dual-frequency, ambiguity resolved, carrier phase differential solution using a Westwind 1124 turbojet aircraft. In 100 approaches, accuracy performance of 0.30 m vertical and 1.6 m horizontal (later corrected to less than 0.50 m horizontal) was demonstrated [65]. Finally, Stanford University fielded a hybrid system using measurements from GPS satellites and low-power, ground-based GPS pseudolites with carrier phase differential processing. The error performance was 0.20 m vertical and 0.40 m horizontal using a Boeing 737 aircraft [66].

In addition to these tests, at least two more LAAS flight test campaigns are reported in the literature. In the first of these, 45 approaches were flown using an FAA Boeing 727 aircraft using multipath-limiting ground station antennas and single-
frequency carrier phase smoothing that was essentially the same as in [11]. The $|\mu| + 2\sigma$
NSE performance achieved was 0.70 m vertical and 0.34 m horizontal. The second of
these test campaigns was performed using a “Beta-LAAS” architecture and a Boeing 737
test aircraft [67]. In this arrangement, three reference receivers were paired with choke
ring antennas with conventional LAAS measurement processing. Error performance was
not described, but the authors concluded that GBAS flexibility is superior to that of the
Instrument Landing System (ILS) based on the various approach configurations tested.

Other dual-frequency civil differential GPS implementations

Against the backdrop of the LAAS program test successes described in the
previous section, a few additional civil differential GPS studies have been published. All
of these incorporate dual-frequency measurements.

One flight test campaign in 1993 successfully implemented a dual-frequency,
carrier phase differential GPS architecture. This study, funded by NASA to develop a
GPS interferometric flight reference system, involved 50 approaches on a NASA Boeing
737 test aircraft [68]. Of these, 17 approaches were selected for analysis, each of which
was completed with an autolanding. During the 4000-ft final approach segment, which
ended at the runway threshold, the vertical $|\mu| + 2\sigma$ error between the GPS system and the
laser tracker truth reference was 0.42 m. The system was designed to demonstrate
accuracy performance for the ambiguity resolved differential carrier phase architecture,
but by intention did not include design elements suitable for commercial use with respect
to integrity, accuracy and continuity.
A notable dual-frequency LAAS architecture was proposed in [69]. After defining the techniques of divergence-free and ionosphere-free smoothing for a local-area differential service, the authors describe a reference station that transmits carrier phase measurements and divergence-free smoothed pseudorange measurements. Several options are provided for the airborne system. These include single frequency processing to provide divergence-free, smoothed L1 pseudoranges or dual-frequency processing to provide divergence-free L1 or L2 pseudoranges. This approach creates a useful framework for CMC techniques and clearly illustrates the cost (in added noise) of removing ionosphere delay altogether from code and carrier measurements. Like the architecture of [69], the DFD1 architecture creates a dual-frequency CMC observable without implementing a Kalman filter, confirms the value of smoothing longer than 100 s to remove multipath oscillations, decouples the smoothing time constants used in reference station and airborne system, and requires continuous carrier phase tracking to realize the architecture’s full benefit. However, the DFD1 architecture smoothes in the position domain rather than in the range domain, and exchanges pseudorange noise and multipath for reducible pseudorange bias instead of filtering the noise and multipath.

A dual-frequency LAAS “toolbox” for ionospheric effects is contained in [70-72]. Besides the helpful additional material on divergence-free and ionosphere-free carrier-code smoothing in [70], these references also contain a LAAS ionosphere monitor architecture that triggers the appropriate smoothing method based on prevailing ionospheric conditions. A VPL is formed to protect against ionospheric error. The authors focus on the tradeoff between smoothing methods and availability, but do not directly
address accuracy improvement in differential GPS systems. Possible dual-frequency (i.e., L1 and L5) upgrades to the WAAS architecture from GPS and Galileo are also discussed in general terms in [73], and an estimate of availability improvements is offered.

**Other single-frequency civil differential GPS implementations**

Two alternate architectures for LAAS-like capability are described in [74] and [75]. The first of these is an apparently single-frequency GPS architecture, coupled with inertial measurements and data from the Russian navigation satellite system GLONASS. This architecture was reported to meet accuracy requirements appropriate for Category IIIb in 36 of 40 precision approaches flown in 1996. Boeing had tested a previous version of this system along with three other differential GPS systems in a 1995 Category IIIb demonstration flight test program that involved 267 approaches using a NASA Boeing 757 aircraft [76]. Similar architectures are envisioned for GPS-Galileo use.

The second alternate architecture, which is described in [75], is LAAS, modified by the addition of a ground-based airport pseudolite. A pseudolite emits a GPS satellite-like signal at relatively close range to the aircraft to obtain better performance than that provided by satellites alone. In 35 precision approaches with a NASA King Air aircraft, this architecture yielded not only $|\mu| + 2\sigma$ error performance of 0.46 m vertical and 0.43 m lateral, but also reduced VPL to less than 2.3 m and LPL to less than 2.0 m.

**Military dual-frequency differential and relative GPS implementations**

The open literature documents several military GPS-based systems with requirements for precision approach and landing. Chief among these are the land and sea-based versions of the Joint Precision Approach and Landing System (JPALS), known as
Local Area Differential GPS (LDGPS) and Shipboard Relative GPS (SRGPS), respectively. Key characteristics of these systems, which are currently under development, are described here as an indication of the capabilities of current dual-frequency, differential GPS technology in a precision approach and landing role.

The planned LDGPS is a LAAS-like differential GPS system that uses carrier-smoothed code observables derived from carrier phase and military P(Y) code phase measurements on both L1 and L2 GPS frequencies. More than 280 precision approaches were flown in 2001 in unjammed and jammed conditions in the project demonstration phase using a U.S. Air Force C-12J (a military aircraft similar to King Air civilian aircraft) [77]. In 2004, range domain smoothing techniques were tested using the same aircraft type in 17 approaches in unjammed, benign conditions [78]. In these later tests, rms errors with single frequency LAAS carrier-code smoothing as defined in [1] were 0.246 m vertical and 0.271 m horizontal. With dual-frequency, divergence-free smoothing using the same 100-s time constants in air and ground, these rms errors were 0.237 m vertical and 0.241 m horizontal. With dual-frequency, ionosphere-free smoothing, the rms errors were 0.766 m vertical and 0.551 m horizontal, larger than for other methods because of the greater number of observables required by the ionosphere-free method.

With tighter performance requirements than LDGPS, the SRGPS architecture is being developed as a relative kinematic carrier phase tracking (RKCPT) differential solution. As detailed in [79-81], the architecture uses a floating integer number of carrier phase cycles during the initial phase of the approach. The integer ambiguity is then fully
resolved during the approach. In 10 approaches by an F/A-18 aircraft ending in automatic touchdowns on the USS Theodore Roosevelt in April 2001, the \( |\mu| + \sigma \) performance over the last mile of the approach was 0.17 m vertical and 0.15 m lateral [79]. The \( |\mu| + 2\sigma \) performance was 0.25 m vertical and 0.24 m lateral. In more recent research to enhance RKCPT performance where the probability of correct fix (PCF) is difficult to achieve, the dual concepts of extra-redundant geometry and probability of almost-fixed (PAF) solution have been studied, with east-north-up rms errors each less than 0.11 m for fixed baseline static ground test data [82]. This research area has also expanded beyond JPALS proper to describe SRGPS for UASs, as in [83–84].

Interoperability between aircraft and reference stations with LDGPS, SRGPS and LAAS equipment has also been discussed in the literature. In one study, the same Rockwell-Collins civil multimode receiver (MMR) aboard a FedEx-operated Boeing 727 aircraft was flown against a Raytheon JPALS ground station at Holloman Air Force Base in New Mexico and then a Raytheon LAAS ground station in Salt Lake City, Utah the following day [85]. Sixteen approaches were flown at each location. The \( |\mu| + 2\sigma \) performance for two typical approaches was provided. These values were 0.78 m and 1.79 m vertical, and 0.77 m and 1.16 m lateral.

A 2005 study of the LDGPS and SRGPS architectural differences sponsored by the U.S. Air Force recommended that the SRGPS architecture be adopted for LDGPS to obtain the lowest cost and integration risk; that common equipment be used for both, particularly the GPS antenna, GPS receiver and anti-jam technology where possible; and that LAAS interoperability for both JPALS variants be postponed until each architecture
is settled [86]. This study also recommended separate hardware suites for JPALS and LAAS at the ground stations, and separate GPS receivers and processing threads in the aircraft, with outputs to pilot guidance systems selected by a switch.

Numerous other studies in the literature address JPALS multipath error, but none employing the CNMP techniques presented here. Similarly, JPALS integrity and availability are also addressed, but discussion of PLs does not extend beyond the LAAS bias-free formulation except to note that group delay biases present in controlled radiation pattern antennas (CRPAs) will negatively affect accuracy and availability.

**Other relative GPS architectures and applications**

High-accuracy, high integrity relative GPS architectures also appear in the literature. Relative navigation is essentially the same as differential, except that the reference station may move, and the geographic location of the reference station need not be precisely known. All of the architectures described in this section employ carrier phase techniques with full ambiguity resolution.

Autonomous airborne refueling (AAR) techniques focused on military UASs is described in [51, 87-90]. Most of these AAR architectures incorporate an inertial input and update rates higher than 1 Hz. Published NSEs are at the dm level, $|\mu| + 2\sigma$. A civil shipboard helicopter landing architecture, which incorporates dual-frequency semi-codeless tracking and an inertial unit, yields low-latency rms errors less than 0.50 m [91]. Autonomous land vehicles trajectory following, which is described in [92], employs a dynamic base real-time kinematic (DRTK) architecture and appears capable of the same levels of accuracy as the helicopter landing architecture. An antenna baseline measuring
system that uses dual-frequency, semi-codeless tracking GPS receivers and a single inertial unit to stabilize sensors distributed over a single aircraft is described in [93]. In flight tests, this system yielded 1 cm accuracy or better (1 sigma) at a 100-Hz update rate.

**Summary of Innovations in the DFD1 Processing Architecture**

When compared to the various differential and relative architectures described in this chapter, the innovations of the DFD1 processing architecture include the following:

- Elimination of the H1 hypothesis and use of the H0 hypothesis via a mid-value select algorithm between three reference antenna-receiver pairs
- Computation of exact composite VPL and HPL values, to a user-specified error tolerance and consistent with univariate normal assumptions
- Option to employ a new class of radial protection levels in two or more dimensions via a quadratic statistical form and a computational method using a series of central chi-square terms; or to employ an efficient approximation of these exact protection levels, along with known error characteristics
- Exchange of pseudorange noise and multipath error for biases that are reduced over time using a modified dual-frequency CNMP algorithm. This algorithm estimates bias bounds from observed maximum and minimum values of the CMC observable, and is employed at the reference station and in the aircraft
- Aircraft-only differential carrier phase position domain smoothing (L1 only), which is divergence free because of its implementation outside the range domain and its use only of the current and previous GPS epochs
Significance of the DFD1 Processing Architecture

Achieved flight test accuracies in the differential and relative GPS architectures described in this chapter are summarized in Table 2 below. The table is divided into two sections by statistic type. The current state of the art in published accuracies of high-accuracy, high-integrity differential and relative GPS applications is readily discernible.

One of the error statistics is for total system error (TSE), which is composed of NSE and flight technical error (FTE), which in turn is defined as the deviation of actual aircraft path from the intended path.

Table 2: Summary of achieved accuracy performance of differential and relative GPS architectures in flight tests

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Application</th>
<th>Details and Method (Year)</th>
<th>Accuracies</th>
<th>Statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>68</td>
<td>Differential</td>
<td>From 17 of 50 approaches in Boeing 737-100, dual-frequency, carrier phase differential (1993)</td>
<td>0.42 m vert, $</td>
<td>\mu</td>
</tr>
<tr>
<td>64</td>
<td>Differential</td>
<td>102 approaches with autoland in Boeing 727-200, single-frequency, C/A code, unfiltered, at 50 ft HAT (1995)</td>
<td>1.30 m vert, 0.90 m lat, $</td>
<td>\mu</td>
</tr>
<tr>
<td>11</td>
<td>Differential</td>
<td>From 50 of 100 approaches including autoland with Boeing 757, single-frequency, C/A code with carrier phase differential velocity, at 50 ft HAT (1994-1995)</td>
<td>1.10 m vert, 1.40 m lat, $</td>
<td>\mu</td>
</tr>
<tr>
<td>65</td>
<td>Differential</td>
<td>100 approaches in Westwind 1124, dual-frequency, ambiguity resolved, carrier phase differential, at 50 ft HAT (1995)</td>
<td>0.30 m vert, 0.50 m lat, $</td>
<td>\mu</td>
</tr>
<tr>
<td>66</td>
<td>Differential</td>
<td>110 approaches with autoland in Boeing 737, carrier phase with low-power pseudolites at 50 ft HAT (1994)</td>
<td>0.20 m vert, 0.40 m lat, $</td>
<td>\mu</td>
</tr>
<tr>
<td>Ref.</td>
<td>Application</td>
<td>Details and Method (Year)</td>
<td>Accuracies</td>
<td>Statistic</td>
</tr>
<tr>
<td>------</td>
<td>------------------------------------</td>
<td>------------------------------------------------------------------------------------------</td>
<td>----------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>67</td>
<td>Differential</td>
<td>45 approaches including autoland in Boeing 727, single-frequency, C/A code with carrier phase differential velocity (1997)</td>
<td>0.70 m vert, 0.34 m lat</td>
<td>$</td>
</tr>
<tr>
<td>75</td>
<td>Differential</td>
<td>35 approaches in King Air, single-frequency, C/A code with carrier phase smoothing and ground-based pseudolite (1997)</td>
<td>0.46 m vert, 0.43 m lat</td>
<td>$</td>
</tr>
<tr>
<td>79</td>
<td>Relative Air-to-Ship</td>
<td>10 approaches in F/A-18 with automatic touchdown on aircraft carrier, relative kinematic carrier phase tracking (2001)</td>
<td>0.25 m vert, 0.15 m lat</td>
<td>$</td>
</tr>
<tr>
<td>51, 87-90</td>
<td>Relative Air-to-Air</td>
<td>Multiple flights, multiple aircraft in refueling scenarios, dual-frequency carrier phase processing aided by P(Y) code and inertial unit (2005-2006)</td>
<td>$&lt; 0.10 \text{ m all dimensions}$</td>
<td>$</td>
</tr>
<tr>
<td>78</td>
<td>Differential</td>
<td>17 approaches in C-12J, single-frequency carrier-smoothed code observables (2004)</td>
<td>0.25 m vert, 0.27 m horz</td>
<td>rms NSE</td>
</tr>
<tr>
<td>78</td>
<td>Differential</td>
<td>17 approaches in C-12J, dual-frequency, divergence-free carrier-smoothed code observables derived from carrier phase and military P(Y) code phase measurements (2004)</td>
<td>0.24 m vert, 0.24 m horz</td>
<td>rms NSE</td>
</tr>
<tr>
<td>78</td>
<td>Differential</td>
<td>17 approaches in C-12J, dual-frequency, ionosphere-free carrier-smoothed code observables derived from carrier phase and military P(Y) code phase measurements (2004)</td>
<td>0.77 m vert, 0.55 m horz</td>
<td>rms NSE</td>
</tr>
<tr>
<td>91</td>
<td>Relative Air-to-Van</td>
<td>Multiple approaches in small helicopter to mobile van test platform, dual-frequency semi-codeless tracking and inertial unit (2005)</td>
<td>$&lt; 0.50 \text{ m radial}$</td>
<td>rms NSE</td>
</tr>
<tr>
<td>93</td>
<td>Relative on Single Aircraft</td>
<td>Antenna baseline measuring system, dual-frequency, semi-codeless tracking GPS receivers, inertial unit to stabilize sensors distributed over single aircraft (2004)</td>
<td>$&lt; 0.01 \text{ m all axes}$</td>
<td>$1\sigma$</td>
</tr>
</tbody>
</table>
The DFD1 processing architecture is significant to the extent that it performs well in three comparisons that directly align with the research questions. First, the protection level methodology used should not only be statistically exact, but also avoid any excess overbounding. Any overbounding that does exist in an approximation to the exact protection level value should be quantified. Second, the architecture should enable a major improvement, as reflected in reduced position errors and tightened protection levels, when compared to conventional LAAS in similar conditions. In this way, the cost to add dual-frequency capability and a revised datalink specification to LAAS may be considered worthwhile. Third, the architecture should be able to approach the $|\mu| + 2\sigma$ NSE accuracy performance of dual-frequency, real-time kinematic carrier phase systems while using C/A code phase and carrier phase processing that does not require integer ambiguity resolution techniques. Granted, other processing-intensive algorithms are introduced with the DFD1 architecture. However, the accuracy results presented in chapter 8 for a single antenna-receiver combination at the reference station may be able to compare favorably to the values listed in Table 2 (most of which are for reference stations with multiple antenna-receiver pairs), while not increasing the burden of processing in the expected operational scenario. To the degree that the DFD1 architecture performs well in these three comparisons, it will constitute a significant step forward for high-accuracy, high-integrity differential and relative GPS applications.
CHAPTER 3: STATISTICAL CHARACTERIZATION OF COMPOSITE PROTECTION LEVELS

This Chapter as Case Study Illustration

This chapter is a case study-type illustration of a general statistical model developed in [94-97], which is applied here to the linearized solution method of the GPS equations. The modeling of GPS pseudoranges as unbiased, uncorrelated, Gaussian random variables is well known and has great practical value. However, this chapter extends this model by redefining the pseudoranges as uncorrelated but biased Gaussian random variables, and then uses the properties of multiple linear regression to define the distributions of the resulting GPS position domain variables \(x, y, z\) and \(t\) as multivariate normal with non-zero mean. The advantages of this novel approach are apparent in the resulting definition of protection levels that are less conservative than those constructed for the unbiased model. Subsequent chapters illustrate potential reductions to the sizes of GPS protection levels, with obvious benefits to GPS system integrity, availability and continuity.

Designation of Random Variables

**The GPS measurement model**

It is assumed that a differential GPS system is operating normally. A linearized GPS measurement model is also assumed. Although this derivation is not limited to LAAS applications, it begins with the same GPS measurement model as defined for LAAS in [1]. The east-north-up locally level coordinate frame assumed here is a slight
modification to [1], in which the LAAS horizontal coordinate axes are either aligned with the runway or are arbitrary.

The GPS measurement model is defined as follows:

\[ \Delta y = H \Delta x + \epsilon \]  

where

- \( \Delta y \) = the \((M\times1)\) vector of ranging measurements, minus expected ranges based on known satellite positions, assumed user position and user clock offset
- \( H \) = the \((M\times4)\) observation matrix, in which the first three elements of each row form a unit vector from the user to the satellite in the locally level coordinate frame, and the fourth element is 1
- \( \Delta x \) = the \((4\times1)\) vector from the user’s assumed position to the user’s true position in the locally level frame, including time as a fourth dimension
- \( \epsilon \) = the \((M\times1)\) vector of ranging measurement errors, to be defined further below

The difference between assumed user position and true user position may be estimated as follows, through the use of multiple linear regression techniques that employ weighted least squares:

\[ \Delta \hat{x} = S \Delta y \]  

where

\[ S = (H^T W H)^{-1} H^T W \]  

Weighting matrix \( W \) is defined as follows:
As stated in the previous chapter, \( \sigma_i^2 \) is the variance of the \( i \)-th satellite measurement’s unbiased, uncorrelated, Gaussian range domain error that results from ground receiver noise and multipath, airborne receiver noise and multipath, and residual delays from ionospheric and tropospheric gradients between the reference station and the user. Matrix \( S \) is the projection matrix from the range domain to the position domain.

The GPS user range domain error vector \( \epsilon \) may now be characterized using \( M \) normally distributed, discrete time random processes:

\[
\mathbf{u}_{i,k} \sim N(\mu_i, \sigma_i^2)
\]

where

- \( u_{i,k} \) = the \( i \)-th random sequence that represents a continuous time random variable, uniformly sampled at period \( \Delta T \) seconds
- \( \mu_i \) = the mean of the \( i \)-th random sequence
- \( \sigma_i^2 \) = the variance of the \( i \)-th random sequence
- \( i = 1, 2, ..., M \) is the satellite index
- \( k = ..., -2, -1, 0, 1, 2 ... \) is the sample index and time variable

These processes are assumed to be discrete-time Gaussian stationary with constant mean and variance over the duration of an aircraft precision approach and

\[
\mathbf{W}^{-1} = \begin{bmatrix}
\sigma_1^2 & 0 & \cdots & 0 \\
0 & \sigma_2^2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \sigma_M^2
\end{bmatrix}
\]

(8)
landing operation. Accordingly, the derivations presented in this section are equally valid for each GPS measurement epoch (i.e., the instant when measurements from all satellites are simultaneously captured). Therefore, only the current epoch, where sample subscript \( k = 0 \), is considered. The sample subscript \( k \) is suppressed.

**Range domain random variables**

Let \( M \)-dimensional real random vector \( \mathbf{u} \), which consists of uncorrelated random variables \( u_1, u_2, \ldots, u_M \), characterize range domain error for the duration of the precision approach and landing operation. It follows from the definition of \( u_{i,k} \) above that each of these random variables is normally distributed with parameters mean \( \mu_i \) and variance \( \sigma_i^2 \). These parameters are themselves constant over the operation, but they may differ from one satellite to another during this time interval.

This definition is a departure from conventional treatment of range domain user measurement error, in which \( u_1, u_2, \ldots, u_M \) are considered uncorrelated, zero mean, identically distributed normal variables. The conventional model has been widely accepted not because it fits GPS behavior perfectly, but rather because it has worked without imposing a great deal of complexity for minimal added performance gain [22]. However, new GPS user requirements are pressing the system to its limits and, as noise filtering is increasingly effective, the presence of biases is receiving greater recognition. Therefore, long-standing assumptions are being reexamined by researchers in this field. This chapter constitutes one such reexamination.
The model proposed in this chapter is motivated by the desire to make practical use of the phenomenon of biases in user GPS measurements that persist essentially unchanged over the five to ten-minute duration of a precision approach, but may change noticeably by the beginning of the next approach. Indeed, GPS range domain measurements are known to exhibit both correlation over time and correlation between simultaneous measurements from different satellites [22].

Several different models may be constructed to describe this phenomenon. The conventional range domain error assumption of zero mean, uncorrelated, identically distributed normal variables is straightforward as noted above. However, this approach inhibits the use of all available information, particularly persistent biases, to set protection levels with minimal overbounding. In another possible set of assumptions, range domain error may instead be considered as a linear process driven by white noise. This alternative has the virtues of constant mean and variance, thus ensuring wide-sense stationarity, and normally distributed error in the range domain. Furthermore, realizations of this process may legitimately take on successive values distant from the mean for an extended period of time, relative to the short duration of a precision approach. This behavior can therefore model a slowly-changing bias, while remaining assured of convergence [98]. However, the ability to translate the current deviation from the constant process mean (i.e., the instantaneous bias) into the position domain, and there to make statistical sense of it, remains out of reach. In a third alternative set of assumptions described in [31], the convolution of uniform and Gaussian distributions represents the
addition of bias-like and noise-like range-domain errors, respectively. This description is straightforward, but does not yield a normal distribution in the position domain.

For all these reasons, the user range error $u$ is defined as normally distributed, but over the limited time horizon of a typical aircraft precision approach and landing operation. Mathematically, this model represents a piecewise fit of a succession of $N(\mu_i, \sigma_i^2)$-distributed random variables over time. Parameters $\mu_i$ and $\sigma_i^2$ for the $i$-th satellite may change from one subdomain, or time segment, to another. However, within the $k$-th subdomain, $\mu_i$ and $\sigma_i^2$ are assumed to be constant. Quantities $\mu_i$ and $\sigma_i^2$ may also be described as piecewise continuous functions, each of which consists of a series of steps. These steps occur on transition from one subdomain to the next. Given the frequent jumps in protection levels seen during a typical precision approach as currently implemented, discontinuities in $\mu_i$ and $\sigma_i^2$ at subdomain boundaries are bounded magnitude do not appear to present a problem to aircraft operations.

As explained in chapter 5, the value of bias $\mu_i$ for a given satellite is not generally known, but may be contained within symmetric bounds that decrease over time. Although the range domain bias values are uncorrelated between the satellites, the signs of the range domain bias bounds are perfectly coordinated between the satellites—that is, they are chosen to conspire—to produce the largest position domain bias bound possible. This worst-case combination is applicable when setting protection levels, but does not come into play when solving for GPS positions. If pre-smoothing with short time constants is performed in the GPS receiver, the assumption on the lack of correlation applies to every
$p$-th GPS epoch, where $p$ is the square root of the inverse of the ratio of the averaging
time to the smoothing time constant. Chapters 4 and 5 contain further details on this
concept.

**Position domain random variables**

The position domain error, defined as the difference between estimated position
offset $\Delta \hat{x}$ and true position offset $\Delta x$ from an assumed position, may be characterized as a
four-dimensional real random vector $q$ with component random variables $x, y, z$ and $t$.
These are east, north and up in a locally-level coordinate system, and user receiver clock
offset from GPS time. Random vector $q$ has a quadvariate normal distribution (i.e.,
multivariate normal with four dimensions), inherited from multivariate normal vector $u$
via linear transformation, as defined below:

$$q = Su \quad \text{or} \quad \begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix} = S \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_M \end{bmatrix}$$

(10)

where projection matrix $S$, defined in Equation (7), above is a deterministic linear
transformation. Then random variable $x$, which is the position domain error in the east
direction, is written as

$$x = \sum_{i=1}^{M} s_{1,i} u_i$$

(11)

where $s_{1,i}$ is the element in the first row and $i$-th column of matrix $S$. It is assumed that the
elements of matrix $S$ are known with certainty. Therefore, any actual geometry errors due
to ephemeris inaccuracies, user position error and the like may be projected onto the line
of sight between the user’s assumed position and a satellite’s calculated position, and
then combined with the pseudorange bias error. Thus, the elements of matrix $S$ may be
treated deterministically as constants instead of as additional random parameters. In
LAAS development, a great deal of effort was devoted to ensuring that large ephemeris
errors would not result in differences in the projection of errors between ground and air.
The reference station should be assigned to monitor this error source.

**Mean and variance of vectors $u$ and $q$**

Instead of deriving the mean vector and variance-covariance matrix of $q$ using full
matrix notation and beginning directly with random variables $x, y, z$ and $t$, as exemplified
in [22] and [99], the derivation provided in this section begins in the range domain. The
position domain error characterization is then obtained using the simple linear
transformations of the GPS measurement model. This alternate derivation intentionally
begins at a basic level in order to illustrate the simplicity of the composite PL approach as
clearly as possible.

Univariate normal random variable $u$, is defined with mean $\mu$ and variance $\sigma^2$:

$$ u \sim N(\mu, \sigma^2) $$

The moment generating function for $u$ is

$$ m(v) = \exp \left( \mu v + \frac{v^2 \sigma^2}{2} \right) $$

The mean and variance of the range domain error are obtained by differentiating this
function and evaluating the results at zero:
The expected value and variance of position domain random variable \( x \) may then be written using the well-known theorem on the numerical characteristics of a linear combination of random variables, which is exemplified by Theorem 5.12 of [100]:

\[
E(x) = \sum_{i=1}^{M} s_{1,i} \mu_i
\]

and

\[
V(x) = \sum_{i=1}^{M} s_{1,i}^2 V(u_i) + 2 \sum_{1 \leq i < j \leq M} s_{1,i} s_{1,j} \text{Cov}(u_i, u_j)
\]

where the double sum is computed over all pairs \((i, j)\) with \(i < j\) using constants \(s\). Since for \(i \neq j\) the covariance of these uncorrelated random variables is always zero, the second term disappears and the variance of \( x \) becomes

\[
V(x) = \sum_{i=1}^{M} s_{1,i}^2 \sigma_i^2
\]

Expected values \( E(y), E(z) \) and \( E(t) \) and variances \( V(y), V(z) \) and \( V(t) \) can be defined by following along the same lines in terms of \( s_{2,i}, s_{3,i} \) and \( s_{4,i}, \) respectively.

The covariance of random variables \( x \) and \( y \) may be also determined from Theorem 5.12 of [100]:

\[
E(u) = \left. \frac{dm(v)}{dv} \right|_{v=0} = \mu
\]

\[
V(u) = \left. \frac{d^2m(v)}{dv^2} \right|_{v=0} = \sigma^2
\]
Since $\text{Cov}(u_i, u_j)$ is simply the variance where $i = j$ and is zero elsewhere,

$$\text{Cov}(x, y) = s_{1,1}s_{2,1}\sigma_1^2 + s_{1,2}s_{2,2}\sigma_2^2 + \ldots + s_{1,M}s_{2,M}\sigma_M^2$$  \hspace{1cm} (20)

Then the variance-covariance matrix is

$$R = \begin{bmatrix} V(x) & \text{Cov}(x, y) & \text{Cov}(x, z) & \text{Cov}(x, t) \\ \text{Cov}(x, y) & V(y) & \text{Cov}(y, z) & \text{Cov}(y, t) \\ \text{Cov}(x, z) & \text{Cov}(y, z) & V(z) & \text{Cov}(z, t) \\ \text{Cov}(x, t) & \text{Cov}(y, t) & \text{Cov}(z, t) & V(t) \end{bmatrix}$$  \hspace{1cm} (21)

where the remaining off-diagonal elements are derived in the same manner as $\text{Cov}(x, y)$.

For the horizontal case, only random variables $x$ and $y$ are of concern, so the corresponding $2 \times 2$ variance-covariance matrix is written as

$$R_{xy} = \begin{bmatrix} V(x) & \text{Cov}(x, y) \\ \text{Cov}(x, y) & V(y) \end{bmatrix}$$  \hspace{1cm} (22)

**Modeling noise propagation using unweighted and weighted least squares**

If noise sigma values are not considered uniform for all the satellites, then the propagation of non-uniform noise must be modeled when constructing the variance-covariance matrix, which in turn is used to set the protection level. Such is the case in this statistical model. This modeling happens automatically in the equation for VPL for the LAAS H0 hypothesis, in which the least squares projection matrix $S$ is formed using weighting terms $1 / \sigma_i^2$ and multiplied by the squares of the corresponding elements of the $S$ matrix. The equation for $VPL_{H0}$ is reproduced here for the purpose of comparison:
If, however, the unweighted least squares method is used for position solutions rather than the weighted method, then multiplying \( \sigma_i^2 \) by the squares of the elements of the unweighted S matrix to create desired terms of variance-covariance matrix \( R \) no longer correctly models the propagation of non-uniform noise. In this case, the revised variance-covariance matrix is instead obtained as described in [12], using notation \( G \) for the unweighted projection matrix:

\[
R' = G Q G^T
\]  

(24)

where the variance-covariance matrix of range domain error \( u \) is

\[
Q = 
\begin{bmatrix}
\sigma_1^2 & 0 & \cdots & 0 \\
0 & \sigma_2^2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \sigma_M^2
\end{bmatrix}
\]

(25)

and

\[
G = (H^T H)^{-1} H^T
\]  

(26)

In this case, the corrected VPL equation would use the (3, 3) element of matrix \( R' \) as follows:

\[
VPL = K_{ffmid} \sqrt{R'_{3,3}}
\]

(27)

If position solutions are obtained using a different weighting scheme than either unweighted least squares or least squares weighted by \( 1 / \sigma_i^2 \), then an additional
correction is required. For the case of least squares weighted by \(1 / \mu_i\), the inverse of the weighting matrix is

\[
W^{-1} = \begin{bmatrix}
\mu_1 & 0 & \cdots & 0 \\
0 & \mu_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \mu_M
\end{bmatrix}
\]  
(28)

and the \(S\) matrix, as before, is defined as

\[
S = (H^T W H)^{-1} H^T W
\]  
(29)

Then the revised covariance matrix that correctly models the propagation of noise through the least squares method weighted by \(1 / \mu_i\) is

\[
R^* = S \Sigma S^T
\]  
(30)

This is the variance-covariance matrix used to create the post-CNMP algorithm protection levels for flight data presented in chapter 8.

**Bias and bias uncertainty**

As will be shown in chapter 5, an estimator \(\hat{B}_i(t)\) for the bias \(\mu_i\) in pseudorange measurements from the \(i\)-th satellite at time \(t\) is developed based on the CMC observable. The notation \(B\), more familiar than \(\mu\) in GPS contexts, is generally used for range domain bias in the remainder of this dissertation. Through the use of CNMP processing, the estimation error decreases over time, as long as continuous GPS carrier phase tracking is maintained for the satellite. The increasingly accurate bias estimate is removed from the pseudorange at each GPS measurement epoch to decrease position error. As visually
illustrated in chapter 4, protection levels are always calculated by taking the worst-case vector combination of the various biases on the satellites.

Estimator $\hat{B}_i(t)$ is assumed to have a uniform sampling distribution [31]. This distribution constitutes a 100% confidence interval for $\hat{B}_i(t)$, centered at zero. The PL is a worst-case formulation, and the PL is applied symmetrically on both sides of the position estimate. Therefore, the absolute value of the minimum or maximum of the distribution, rather than the value of $\hat{B}_i(t)$ itself, is used to determine the PL. Whether the maximum or minimum is used does not matter, since the interval is centered about zero and the absolute value of each is the same. This value is determined analytically for a particular differential GPS installation, as described in chapter 5.

**Statistical Method for Exact VPL$_c$ and HPL$_c$**

When viewed within the statistical framework described above, the original VPL$_c$ expression is actually a conservative approximation for a true value that can be derived statistically. This true value $\rho$ is the magnitude of the upper and lower integration limits, which are symmetric about zero, of the integral of the probability density function of a normal distribution having a non-zero mean and a given probability $p$. Consider such a distribution with mean $B$ and standard deviation $\sigma$, defined as follows:

$$p = \int_{-\rho}^{\rho} \frac{1}{\sigma \sqrt{2\pi}} \exp\left\{-\frac{(r-B)^2}{2\sigma^2}\right\} dr$$

This integral is also illustrated in Figure 5:
The original VPLc value, computed as $B + K\sigma$ and shown in Figure 5, is an artifact of the protection level for the zero mean distribution from which it was derived. Because the PL must be symmetric about the origin, the integral of the probability density function with integration limits $-(B + K\sigma)$ and $B + K\sigma$ contains a significantly larger probability content than for the original integration limits $-K\sigma$ and $K\sigma$. Note that the interval $[-K\sigma, K\sigma]$ for the zero-mean (i.e., unbiased) case is the same size as the interval $[B - K\sigma, B + K\sigma]$ for the biased case, even though it has been translated from the origin. This translated interval, however, cannot be used as a protection level due to its asymmetry about the origin. If instead the value of $\rho$ can be found numerically by varying the integration limits until the desired value of $p$ is achieved, then an exact probability may be enclosed by a new VPLc, to the specified error tolerance and consistent with the assumptions. The new exact protection level is hereafter referred to as VPLc, and the original protection level, which is an approximation, is termed VPLc1.
The same approach is applied to define $HPL_c$. Similarly, its approximation will be referred to as $HPL_{c1}$. The quantity $HPL_c$, like the original HPL, is defined under a univariate normal distribution as the projection of horizontal errors onto the major axis of the error ellipse in the horizontal plane. Because the semi-major axis length of an ellipse is its largest radius, the HPL and $HPL_c$ are worst-case measures of generic lateral, or cross-track, error, and are used when the orientation of a runway is unknown.

The HPL is not properly a radial measure, in which a specified probability is enclosed within a given radius. In fact, an HPL interpreted as a radial bound will always contain a smaller probability than the same HPL value interpreted as a linear bound. Figure 6 illustrates the errors projected onto the “linear” HPL but not enclosed within the “radial” HPL. These points between the straight line and circle are colored light blue, and are shown for the $[B - 3\sigma, B + 3\sigma]$ case for illustration purposes. As the sigma multiplier $K$ increases, the probability of error in this region decreases. Conversely, a radial protection level that contains the same probability content as a linear protection level will always be a little larger.
Radial protection levels are not defined for aircraft precision approach, simply because an airport is a world full of rectangles, right angles, straight lines and, geometrically speaking, planes. However, the radial protection level concept may be of significant value in other airborne operations, particularly in relative navigation applications such as UASs or spacecraft flying in close formation. Building on the characterization of position domain error as multivariate normal with non-zero mean, the following section defines a new class of radial protection levels for use in such applications.
Use of Quadratic Form for $K$-dimensional Radial Protection Levels

A statistical quadratic form can be used to implement a radial integration limit for probability content under a multivariate normal distribution. As will be seen, this is a major benefit from the assumption that position domain error is normally distributed.

In [94] and [95], Ruben derived an infinite linear combination representation for the probability density function of a quadratic form for a finite number of correlated normally distributed random variables. He also provided estimates of error and a proof of convergence. This derivation is applicable to the present case, since the position domain random variables $x, y, z$ and $t$ are correlated in general by transformation through $S$, even though the underlying range domain random variables $u_1, u_2, \ldots, u_M$ are assumed to be uncorrelated. What this means practically is that the probability content of a $K$-dimensional fixed ellipsoid $R^*$ under a multivariate normal distribution can be evaluated numerically to an arbitrary level of accuracy, regardless of the ellipsoid’s size, orientation and offset from the distribution’s center. A sphere, circle and line—already the basis of existing GPS error characterizations—are all special cases of the ellipsoid $R^*$. More importantly, the ellipsoidal integration volume enables the use of new, more flexible error characterizations for evaluating protection levels in GPS.

The terms of Ruben’s infinite series are chi-square distribution functions, each with a scale parameter of arbitrary size. Kotz et al. reformulated this derivation in [96], incorporating Ruben’s work as a special case along with several similar expansions, then added estimates of error and proof of convergence for the reformulation. Sheil and
O’Muircheartaigh derived an efficient algorithm in [97] for Ruben and Kotz’s central chi-square expansion.

The derivation appearing below closely follows that in [97]. For the sake of generality, the derivation is carried out for the general, $K$-dimensional case.

**Central Chi-square Representation of Quadratic Form**

Suppose that a $K$-dimensional random vector $q$ has a multivariate normal distribution with mean vector $\mu$ and variance-covariance matrix $R$, which is assumed to be non-degenerate. Individual elements of $q$ are correlated.

The quadratic form $(q + a)^T C (q + a)$ in effect establishes an ellipsoidal region of integration, centered at point $a$, and with shape and orientation determined by the symmetric positive definite, or semi-definite, square matrix $C$. For $K = 3$, defining the matrix $C$ as the $3 \times 3$ identity matrix establishes a spherical region of integration. Similarly, a circular region of integration is established by defining matrix $C$ as the $2 \times 2$ identity matrix. Reduced rank realizations of matrix $C$ may also be defined if protection levels are desired for all $K$ dimensions.

Linear transformations are performed as follows:

$$q - \beta = L^T v, \quad a + \beta = L^T b$$

so that the individual elements of vector $v$ are uncorrelated and therefore independent, identically distributed random variables with the common standard normal distribution. Matrix $L$ is the upper triangular matrix derived from the variance-covariance matrix $R = L^T L$. The eigenvalue decomposition is then applied to matrix $LCL^T$ to
produce matrix $\mathbf{A} = \text{diag}(\alpha_i)$, whose main diagonal elements are the eigenvalues $\alpha_i$; and matrix $\mathbf{D}$, whose columns are the eigenvectors.

In [97] it is shown that

$$P \{(\mathbf{q} + \mathbf{a})^T \mathbf{C} (\mathbf{q} + \mathbf{a}) \leq \tau\} = P \{(\mathbf{v} + \mathbf{b})^T \mathbf{A} (\mathbf{v} + \mathbf{b}) \leq \tau\}$$

(33)

Here the value of scalar $\tau = r^2$, where $r$ is the radius of the region of integration.

Let $f(m, w)$ be the probability density of a central chi-square distributed random variable having $m$ degrees of freedom. Similarly, let $F(m, w)$ be the corresponding cumulative distribution function. Combining Equation (1.13) from [94] and Equation (93) from [96], the probability of the quadratic forms shown in the previous equation may be represented as follows:

$$P \{(\mathbf{q} + \mathbf{a})^T \mathbf{C} (\mathbf{q} + \mathbf{a}) \leq \tau\} = \begin{cases} \sum_{k=0}^{\infty} c_k F(m + 2k, \tau / \zeta), & \text{if } \tau > 0 \\ 0, & \text{if } \tau \leq 0 \end{cases}$$

(34)

Here $m$ is also the rank of matrix $\mathbf{C}$, and $\zeta$ is a distribution-free constant. As in [97], this series is uniformly absolutely convergent for all $\tau > 0$ as long as $0 < \zeta < 2 \min(\alpha_i)$.

In the present implementation, the value of $\zeta = 29 / 32$ is borrowed from [97].

Thus, the series is a mixture representation such that all coefficients $c_k$ are greater than zero and sum to one from $k = 0$ to infinity. Because the $\chi^2$ functions are monotonically decreasing, truncation error after $L$ terms is bounded according to the following expression:
For the computation of the expansion of the series coefficients and $\chi^2$ distribution functions, the reader is referred to [97] and to a Matlab implementation of this algorithm by Genz in [101].

**Using Noise and Conspiring Bias to Set Radial Protection Levels**

$RPL_c$

The worst-case combination of range domain bias and noise is used to form a two-dimensional composite radial protection level ($RPL_c$) for a given probability $P_{pl}$. The range domain bias bound for the $i$-th satellite, $B_{max,i}$, which always has a positive value, is used. The scalar value $RPL_c$ is computed using the following steps:

1. The 2-dimensional position domain variance-covariance matrix $R_{xy}$ is constructed as described above.

2. An $M$-dimensional range domain bias vector $B_{max}$ is created with elements $B_{max,i}$ and by assigning a positive or negative sign to each element. Initially, all of the signs of $B_{max}$ are positive. This signed vector is mapped into the position domain to obtain the bias vector $\beta$ in the x-y plane:

$$\beta = S B_{max}$$  \hspace{1cm} (36)

Although the largest magnitude of $\beta$ with respect to any of the position domain axes may be easily obtained by assigning the same signs to $B_{max}$ as those found in the corresponding row of $S$ (effectively taking the absolute value of each), the
value of $\mathbf{b}$ that produces the largest sum of bias and noise between the axes may be found only through a search that systematically varies the signs of the elements of vector $\mathbf{b}$ while maximizing $r$. If the elements of $\mathbf{B}_{\text{max}}$ are all the same, the orientations of the bias cloud and noise ellipse are correlated, and the search may possibly be shortened by adding logical conditions. Otherwise, options for limiting the search while obtaining the minimum RPL$_c$ that bounds the desired probability are few.

3. The $\chi^2$ series representation previously described is evaluated using quantities $\mathbf{b}$, $\mathbf{R}_{\text{xy}}$, and $P_{pl}$ from the preceding steps, along with dimension $K = 2$, ellipsoid shape matrix $\mathbf{C}$ equal to a (2x2) identity matrix, maximum error in probability $\gamma$ specified by the user (must be smaller than $1 - P_{pl}$), and initial protection level radius $r = 1$ m. This step yields the probability $P_c$ for the given $r$ value.

4. Step 3 is iterated while varying $r$. First, $r$ is changed either in the increment of +0.1 if $P_c < P_{pl}$ or in the increment of −0.1 if $P_c > P_{pl}$, until the difference $P_c - P_{pl}$ changes sign. Then this iteration is repeated using successively smaller increments of $r = (0.1)^j$, $j = 3, 5, 7, 9$. This iterative approach is adapted from [102]. After all iterations are complete, the resulting value of $r$ obtained for the given $P_{pl}$ is stored, along with the signs of the elements of $\mathbf{B}_{\text{max}}$ that produced it.

5. The sign of one element of $\mathbf{B}_{\text{max}}$ is reversed, and steps 2 through 4 are repeated. The pattern of sign changes is such that all possible combinations of signs of $\mathbf{B}_{\text{max}}$ elements are obtained. If the current value of $r$ is larger than the previously
stored maximum value of \( r \), the maximum value is replaced with the current value, along with the sign combination that produced it.

6. When step 5 is completed for all sign combinations of the elements of \( \mathbf{B}_{\text{max}} \), protection level \( RPL_c \) is set to the largest value observed for \( r \) during step 5.

It is clear that this procedure obtains the minimum \( RPL_c \) value that bounds the combination of conspiring range domain biases that, when mapped into the horizontal plane, creates a worst-case offset for any position domain noise ellipsoid. Thus, \( RPL_c \) value is the lower error bound for all simpler but more conservative approximations, because it takes no simplifying shortcuts. The \( RPL_c \) value is exact in the sense that it corresponds to probability value \( P_{pl} \) with arbitrarily small probability error tolerance \( \pm \gamma \), both of which are determined by the user. At the same time, the \( RPL_c \) technique produces an upper bound on radial horizontal error that is consistent with assumptions made regarding the distribution of noise vector \( \mathbf{u} \) and the validity of the bias bound \( \mathbf{B}_{\text{max}} \).

Because of the nested loops implied in steps 3 through 5 above, this procedure’s computational inefficiency may make real-time implementation a challenge. The current Matlab code uses vectorized calculations generally. However, it has not been optimized for execution speed. For example, it calculates \( RPL_c \) for a seven-satellite combination in 18 s on a 3 GHz P4 processor for the fairly benign conditions described in the next chapter. For the more typical constellations with eight to eleven satellites, execution time becomes even more prohibitive. The value of this technique is primarily as a baseline to evaluate the accuracy of more efficient approximations. An approximation to \( RPL_c \) is defined and illustrated by simulation in chapter 4.
The quadratic form described above may be adapted to compute a composite spherical protection level simply by setting the dimension to \( M = 3 \), using a \((3 \times 3)\) identity matrix for \( C \), and using the first three rows and columns of matrices \( R \) and \( S \). Alternately, one can employ the original dimensions \( M = 4 \), leave the other matrices as they are, and use a reduced rank version of \( C \). This positive semi-definite matrix, obviously still symmetric, is shown as follows:

\[
C = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

(37)

A composite ellipsoidal protection level may likewise be computed using a slightly different \( C \). For example, to double the safety margin in the vertical dimension while leaving the horizontal dimensions unchanged, one could use the following matrix:

\[
C = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

(38)

This technique may also be used to vary the shape of the EPL in along-track or cross-track dimensions, or both, depending on the application. If precise time synchronization is essential, the fourth dimension of \( C \) might be used as well. Precise aerial mapping by UASs flying in close formation over populated areas is one example application that might benefit from such a reexamination of protection levels beyond VPL and HPL.
Calculating VPL<sub>c</sub> and HPL<sub>c</sub>

The procedures for obtaining VPL<sub>c</sub> and HPL<sub>c</sub> are similar to those for the radial protection levels, with steps for computing VPL<sub>c</sub> included below as an example. Once the semi-major axis length and angle are found for the horizontal plane, these same steps yield HPL<sub>c</sub>. Since both bias and noise errors map directly either to the vertical axis or the rotated horizontal axis, the determination of biases requires no search. As mentioned previously, the probability integral is performed in one dimension. The steps for VPL<sub>c</sub> are as follows:

1. Total system sigma values for each satellite in the (Mx1) row vector \( \mathbf{\Sigma} \) are mapped onto the vertical axis by multiplying \( \mathbf{\Sigma} \) by the third row of the matrix \( \mathbf{S} \), which corresponds to the vertical dimension. No variance-covariance matrix is needed:

   \[
   \sigma_z = \mathbf{S}_{3,1:M} \mathbf{\Sigma}
   \] 

   (39)

2. The \( M \)-dimensional range domain bias vector \( \mathbf{B}_{\text{max}} \) is created from elements \( B_{\text{max},i} \) as in the RPL<sub>c</sub> procedure, except that the signs of \( \mathbf{B}_{\text{max}} \) are equated to the corresponding signs in the third row of \( \mathbf{S} \), which is for the vertical dimension \( z \). As before, this has the effect of taking the absolute value because positive signs are multiplied by positive signs, and negative signs by negative signs. The vector \( \mathbf{B}_{\text{max}} \) is mapped onto the vertical axis by multiplying by the third row of \( \mathbf{S} \), as in the previous step:

   \[
   \beta_z = \mathbf{S}_{3,1:M} \mathbf{B}_{\text{max}}
   \] 

   (40)
3. Next, the inverse function of the cumulative distribution function of the standard normal distribution is used with input arguments $\sigma_z, \beta_z$ and $P_{pl}$ as in Equation (31). This involves use of a preexisting numerical routine such as the Matlab functions `norminv` or `erf`.

4. Finally, the integration limit $\lambda$ obtained in the previous step, as defined in Equation (31), is assigned as the value of $VPL_c$.

Approximations to these exact PLs are quite useful in implementation. The following chapter provides approximations to $VPL_c$, $HPL_c$ and $RPL_c$, and illustrates them via simulation.
A Computationally Efficient Approximation to $VPL_c$

A computationally efficient approximation to $VPL_c$ already exists in the original $VPL_c$ expression in [5] and in $VPL_{H1}$ as described in [1]. The primary difference between $VPL_{H1}$ and $VPL_c$ as defined in [5] and in this section is that the bounds $B_{\text{max},i}$ are used rather than observed value $B_{i,j}$. This is because the maximum conspiring bias vector sum, relative to the origin, is desired, rather than the observed bias value relative to the average bias computed simultaneously between a set of reference receivers. Since the proposed differential GPS architecture is for a single reference receiver alongside two others in a mid-value select scheme, the reference receiver index $j$ used in [1] is suppressed. This approximation is updated as follows:

$$VPL_{c1} = K_{\text{find}} \sqrt{\sum_{i=1}^{M} s_{3,i}^2 \sigma_{i}^2} + \sum_{i=1}^{M} s_{3,i} B_{\text{max},i}$$  \hspace{1cm} (41)$$

The first term of this expression consists of the fault-free multiplier $K_{\text{find}}$ multiplied by the vertical sigma, which is the square root of element (3, 3) of position domain correlation matrix $\mathbf{R}$. This approximation is illustrated in the x-z plane in Figure 7 for a sample noise ellipse offset by a bias from a 6-satellite simulated combination. The position domain bias components, mapped in the worst-case way from biases of 0.2 m uniform magnitude on each satellite, are shown in solid white, stretching “head-to-tail” from the origin to the center of the ellipse. Thus, this ellipse has been displaced from the origin in the vertical dimension as far as it can be. The projection of this vector sum onto the vertical axis is shown by the dashed red line segment. The length of the noise vector
projected onto the vertical axis, as shown by the dashed blue line segment, is obtained by multiplying the square root of element (3, 3) of variance-covariance matrix $R$ by $K_{ffmd}$.

For this simulated example, $P_{pl} = 0.99999$, which corresponds to $K_{ffmd} = 4.4172$. The value of $VPL_{c1} = 0.78 \text{ m} + 2.19 \text{ m} = 2.97 \text{ m}$ is shown by the horizontal dashed lines and by the sum of the two line segments aligned on the vertical axis.

Figure 7: $VPL_{c1}$ in x-z plane (dashed black bold), with head-to-tail bias vectors (white solid), and bias and noise projected on vertical axis (dashed red, blue)

Since the noise apart from the bias is still characterized completely by the multiplier $K_{ffmd}$, which is retained here without modification, this construction is an
overbounding approximation. This is because the bias error is always less than or equal to the worst-case bias vector sum. In addition, the analogous form of the \( LPL_{H1} \) in [1] is an overbounding approximation for the same reason.

**Two Computationally Efficient Approximations to HPL\(_c\)**

No HPL\(_c\) approximations implemented with a bias bound, rather than a bias value, were found in the literature. However, the form of HPL\(_{H1}\), as specified in [1], can be modified slightly to serve this purpose. A second approximation was obtained from prior unpublished research at Ohio University [103].

**HPL\(_c\) Approximation 1**

The LAAS positioning service HPL specified in [1] for the single-fault scenario, HPL\(_{H1}\), may be adapted to serve as an approximation to HPL\(_c\). As before, bound \( B_{\text{max},i} \) is substituted for observed value \( B_i \), and reference receiver subscript \( j \) is suppressed. Additionally, the absolute value is taken of all the components of projection matrix \( S \).

This is necessary to ensure a worst-case combination without the requirement of a search. This absolute value operator is necessary because of the four-dimensional least squares definition of matrix \( S \). Specifically, if all \( B_{\text{max},i} \) are identical, computing the sum by multiplying by \( S \) without taking the absolute value results in projecting all error into the clock dimension, thereby leaving the resultant bias vector in the horizontal plane unusable with zero length. The revised form is

\[
HPL_{c1} = K_{\text{find}} d_{\text{major}} + \sqrt{\left( \sum_{i=1}^{M} s_{1,i} \cdot B_{\text{max},i} \right)^2 + \left( \sum_{i=1}^{M} s_{2,i} \cdot B_{\text{max},i} \right)^2} \tag{42}
\]
The component bias vectors are summed head-to-tail in the worst-case scenario, and the resultant vector is the position domain bias in the horizontal plane. The remaining component of $HPL_c$ is approximated by the semi-major axis of the 1-sigma noise ellipse, multiplied by $K_{ffmd}$ as in [1]. This combination is illustrated in Figure 8, followed by close-up views in Figure 9 and Figure 10.

![Figure 8: Illustration of $HPL_{c1}$ in horizontal plane, with head-to-tail bias vectors added collinearly to ellipse’s semi-major axis length](image)

In the close-up view of the bias vector combinations of Figure 9, the white head-to-tail vector combination is the worst-case physically realizable bias combination obtained by search, as described in the previous chapter. The black vector combination is
worst-case bias set produced by HPL Approximation 1 (i.e., HPL$_{c1}$). The red dashed line is the resultant of the vectors of Approximation 1, here arranged collinearly with the noise ellipse semi-major axis to illustrate the protection level computation.

Figure 9: Close-up of worst-case horizontal bias combinations: true (white), Approximation 1 (black), and as magnitude (dashed red) in protection level

As indicated above, this approximation technique strips out all sign information and thus allows vector orientations beyond the “range of motion” actually permitted by the projection matrix. This is because the sign combinations of rows one and two of the projection matrix (i.e., for the x and y axes that define the horizontal) are generally different. Therefore, one can set the signs of bias vector $\mathbf{B}_{\text{max}}$ to match only one of them at a time to perform the absolute value operation. By artificially taking the absolute values of both rows at the same time and ignoring opposite signs for the same satellite,
one is in effect asserting that a given satellite can lie on reciprocal bearings from the user at the same time. However, the technique is clearly an overbound because the length of the new resultant bias vector must always be greater than or equal to the original one. Moreover, a search is avoided. In this figure, the black vector combination is shown in its third-quadrant version, which is formed by making all x values positive and all y values negative. Of course, the resultant length is the same in all four quadrants, and the value of HPL\textsubscript{c1} remains the same.

\textbf{HPL\textsubscript{c} Approximation 2}

The second approximation to HPL\textsubscript{c} is essentially a simpler version of the first. The physical limits on vector orientation imposed by projection matrix S are further disregarded by “stretching” the bias vector combination until all the components are collinear. The following form is used:

\[ HPL_{c2} = K \text{\textsubscript{f}} \text{\textsubscript{find}} d_{\text{major}} + \sum_{i=1}^{M} \sqrt{\left(s_{1,i} B_{\text{max},i}\right)^2 + \left(s_{2,i} B_{\text{max},i}\right)^2} \]  

(43)

If the bias bounds for each satellite all have the same magnitude, the form is even simpler:

\[ HPL_{c2} = K \text{\textsubscript{f}} \text{\textsubscript{find}} d_{\text{major}} + B_{\text{max}} \sum_{i=1}^{M} \sqrt{s_{1,i}^2 + s_{2,i}^2} \]  

(44)

The second term in this simplified expression is similar to the form of the GPS Horizontal Dilution of Precision (HDOP), but is not the same. In the HDOP form, the summation operator is inside the square root operator, and applies to each term separately.
Approximation 2 is illustrated by the red dashed line in Figure 10. This approximation is slightly more conservative than Approximation 1, with a larger bias term, and is a simpler calculation. It is also demonstrably an overbound due to the same geometric arguments.

![Figure 10: Close-up of worst-case horizontal bias combinations: true (white), Approximation 1 (black) and Approximation 2 (red dashed)](image)

**Figure 10: Close-up of worst-case horizontal bias combinations: true (white), Approximation 1 (black) and Approximation 2 (red dashed)**

**An Approximation to RPL<sub>c</sub>**

A reasonably computationally efficient approximation to RPL<sub>c</sub> may be obtained by eliminating the full conspiring bias search and its evaluation of the probability integral at each iteration. Instead, the maximum bias length is derived using a less complex search that avoids performing the integral at each iteration. Array processing can considerably speed this reduced search.
For a maximum of $M = 12$ twelve satellites, the number of bias sign combinations $L = 2^M = 4096$. A 12-bit shift register or a $(4096 \times 12)$ array of all the sign combinations may be used to form the sign combinations, which conveniently appear in the same sequence as the standard binary number codes from 1 to $L$. The resultant combinations in the x-y plane, either in scalar or vector form, can be produced using array multiplication. The maximum length found among the resultant vectors in the x-y plane is selected as the worst-case horizontal bias, and added collinearly to the 1-sigma error ellipse semi-major axis, multiplied by $K_{\text{ffmd}}$. This procedure may be summarized as follows:

$$RPL_{c1} = K_{\text{ffmd}} d_{\text{major}} + \max_{i} \left\{ \frac{s^2}{s_{1,i}^2 + s_{2,i}^2} \right\}, \quad i = 1, 2, \ldots, L$$

(45)

This approximation is not guaranteed to be an overbound apart from further analysis. As illustrated previously in Figure 6, use of a univariate sigma value in the definition of a radial measure creates an underbound. However, the collinear geometric arrangement of the maximum bias combination with the 1-sigma semi-major axis, multiplied by $K_{\text{ffmd}}$, reverses this effect to some degree. An analysis of the bounding qualities of this approximation is offered for future research.

**Summary of Protection Level Equations**

The equations for the LAAS baseline PLs and the approximations to the composite PLs for the H0 hypothesis are shown in Table 3. Weighting the least-squares position and protection level calculations by the inverse of variance, as performed in LAAS, could be applied to the composite protection levels. However, from [12] it is observed that the unweighted least-squares method is superior where significant biases
exist. Where biases are several times that of noise sigmas, as is the case after execution of the CNMP algorithm, weighting by the inverse of bias results in an even better fit.

### Table 3: Equations for LAAS and composite protection level approximations

<table>
<thead>
<tr>
<th>LAAS Protection Levels</th>
<th>Approximations to Composite Protection Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>( VPL_{10} ) = ( K_{\text{find}} \sqrt{\sum_{i=1}^{M} s_{3,i}^2 \sigma_i^2} )</td>
<td>( VPL_{c1} = K_{\text{find}} \sqrt{\sum_{i=1}^{M} s_{3,i}^2 \sigma_i^2 + \sum_{i=1}^{M}</td>
</tr>
<tr>
<td>( HPL_{10} ) = ( K_{\text{find}} d_{\text{major}} )</td>
<td>( HPL_{c1} = K_{\text{find}} d_{\text{major}} + \sqrt{\left( \sum_{i=1}^{M}</td>
</tr>
<tr>
<td>( HPL_{c2} ) = ( K_{\text{find}} d_{\text{major}} + \sum_{i=1}^{M} \left(</td>
<td>s_{1,i} B_{\max,i}</td>
</tr>
<tr>
<td>( W = \text{diag}(1/\sigma_i^2) )</td>
<td>( W = \text{diag}(1/ B_{\max,i}) ) or ( W = \text{diag}(1_i) )</td>
</tr>
</tbody>
</table>

### Monte-Carlo Simulation Results

In this section, the results of a Monte-Carlo simulation are provided to illustrate the exact PLs and their approximations. The primary purpose of this section is to help the reader visualize these PLs geometrically and appreciate their merits relative to each other. Although error is quantified, it serves as an illustration rather than an error analysis of these techniques. A full listing of the Matlab code used here is provided in Appendix A.

**Simulation parameters**

The Monte-Carlo simulation was performed in Matlab for a single GPS epoch, with the parameters shown in Table 4. Results for all of the one and two-dimensional PLs and their approximations are obtained for the same simulation run. The error plots shown
previously in Figure 6 though Figure 10 were produced using this simulation code. However, several parameters in those illustrations differ from the values used here.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Value(s)</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Satellite geometry</td>
<td>Chosen from the 24-slot constellation, two hours into GPS day [104]</td>
<td>One of seven satellites available is removed to illustrate a more problematic geometry with six satellites</td>
</tr>
<tr>
<td>User location</td>
<td>30 deg N, 90 deg W</td>
<td>New Orleans</td>
</tr>
<tr>
<td>Number of iterations in the simulation, $N_{\text{sim}}$</td>
<td>100,000</td>
<td>Maximum number that Pentium 4 processor with 2 GB memory can handle in this Matlab implementation</td>
</tr>
<tr>
<td>Probability of hazardously misleading information, $P_{\text{HMI}}$</td>
<td>$1 \times 10^{-4}$, corresponding to $K_{\text{find}} = 3.8906$</td>
<td>Set so that $P_{\text{HMI}} \times N_{\text{sim}} &gt; 1$, to allow a few outliers outside PL in the simulation</td>
</tr>
<tr>
<td>Maximum probability error tolerance, $\gamma$</td>
<td>$1 \times 10^{-9}$</td>
<td>Must be smaller than $P_{\text{HMI}}$, but this very small value is used to ensure an accurate estimate of probability contained by estimated PLs.</td>
</tr>
<tr>
<td>Maximum range domain bias, $B_{\text{max}}$</td>
<td>20 cm</td>
<td>Applied equally to all satellites as an upper bound</td>
</tr>
<tr>
<td>Noise, $\sigma_i$</td>
<td>$\sigma_i^2 = (\sigma_{\text{pr}<em>\text{gnd}}^2 + \sigma</em>{\text{air}}^2 + \sigma_{\text{iono}}^2 + \sigma_{\text{tropo}}^2)$ as in [1]</td>
<td>These terms assumed to be Gaussian noise only. In reality, using them “as is” alongside bias terms might result in some double-counting of bias effects that may be included. Flight test results use a different form</td>
</tr>
<tr>
<td>Ground noise and multipath, $\sigma_{\text{pr}_\text{gnd}}$</td>
<td>From GAD-C4 curve in [1]</td>
<td>Includes thermal noise, multipath, etc.</td>
</tr>
<tr>
<td>Airborne noise and multipath, $\sigma_{\text{air}}$</td>
<td>Noise from AAD-B curve; multipath from AAD-A curve in [1]</td>
<td>AAD-B curve for multipath is still under development, so AAD-A curve is used</td>
</tr>
<tr>
<td>Quantity</td>
<td>Value(s)</td>
<td>Remarks</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>----------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Residual ionospheric delay, $\sigma_{\text{iono}}$</td>
<td>0.02 m</td>
<td>Assuming dual-frequency receivers remove first-order ionosphere effects, this is for higher order iono terms</td>
</tr>
<tr>
<td>Residual tropospheric error, $\sigma_{\text{tropo}}$</td>
<td>Zero</td>
<td>As recommended in App. F of [1] for availability analyses</td>
</tr>
<tr>
<td>Least-squares weighting technique</td>
<td>Unweighted</td>
<td>Best available technique for biases and noise sigmas of comparable size. This is not the case for post-CNMP bias and noise.</td>
</tr>
<tr>
<td>Smoothing</td>
<td>None</td>
<td>Simulation illustrates PLs, not smoothing</td>
</tr>
<tr>
<td>Vertical Dilution of Precision (VDOP)</td>
<td>1.762</td>
<td>Using unweighted LS</td>
</tr>
<tr>
<td>Horizontal Dilution of Precision (HDOP)</td>
<td>1.942</td>
<td>Using unweighted LS</td>
</tr>
</tbody>
</table>

The 1-sigma values of the components of range domain noise $\sigma_i$ prior to multiplication by $K_{\text{find}}$ are shown below in Figure 11, as a function of elevation. The root-sum-square total value of the noise components is shown as well. The 20 cm value of range-domain bias bound $\mu_{\text{max}}$ is approximately the same as the 1-sigma value for total noise error for higher satellite elevation values.
The position domain error of this simulation in x, y and z axes appears in Figure 12. Since the purpose of this simulation is to illustrate protection levels, no smoothing has been performed. Similar plots of flight data NSE, processed by the CNMP and CPDS algorithms, are provided in the flight test results in chapter 8.
**VPL\(_c\) and its approximation**

The error ellipse projected onto the x-z plane, with exact and approximate VPL\(_c\) values shown as horizontal lines, appears in Figure 13. The vertical bias of 1.083 m is apparent in the upward shift of the error ellipse—again the worst case vertical displacement possible for this epoch. A confidence interval was not determined for either VPL\(_c\) in this illustration, but several runs of the simulation placed 99.99 ± 0.01 % of the errors inside the exact protection level value of 7.661 m.
Figure 13: Simulated biased, unsmoothed error in x-z plane, projected onto vertical axis with ±VPLc (solid lines) and one approximation to VPLc

The performance of VPLc and its approximation in simulation for this single GPS epoch is shown in Table 5 below:

<table>
<thead>
<tr>
<th>VPLc Method</th>
<th>Value in m (99.99%)</th>
<th>Percent of simulated errors enclosed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact</td>
<td>7.661</td>
<td>99.992%</td>
</tr>
<tr>
<td>Approximation 1</td>
<td>7.962</td>
<td>99.997%</td>
</tr>
</tbody>
</table>
**HPL_c and its approximations**

Similarly, errors projected onto the horizontal x-y plane and then onto the rotated horizontal axis, along with HPL_c and its two approximations, are shown in Figure 14. The angle between the x axis and the rotated horizontal axis is – 41 degrees. The set of biases shown is the one that produced maximum displacement when projected onto the rotated horizontal axis.

![Figure 14: Simulated biased, unsmoothed error in x-y plane, projected onto rotated axis with ± HPL_c (solid lines) and two approximations to HPL_c](image)

The simulated performance of HPL_c and its two approximations appears below in Table 6:
Table 6: Simulated performance of HPL<sub>c</sub> methods

<table>
<thead>
<tr>
<th>HPL&lt;sub&gt;c&lt;/sub&gt; Method</th>
<th>Value in m (99.99%)</th>
<th>Percent of simulated errors enclosed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact</td>
<td>4.683</td>
<td>99.991%</td>
</tr>
<tr>
<td>Approximation 1</td>
<td>4.968</td>
<td>99.996%</td>
</tr>
<tr>
<td>Approximation 2</td>
<td>5.049</td>
<td>99.996%</td>
</tr>
</tbody>
</table>

RPL<sub>c</sub> and its approximation

The set of conspiring position domain bias combinations available in the horizontal plane appears in Figure 15. The bias cloud exhibits symmetry through the origin, as one might expect for positive and negative pairs of bias combinations. The angle between the x axis and the approximate “major axis” of the bias cloud is approximately –25 degrees, sixteen degrees offset from –41 degree angle of the noise ellipse’s major axis. In general, until the noise ellipse is considered, it is not known which of these biases will produce the largest protection level radius. Because the bias cloud and noise error ellipse in the horizontal plane do not have the same orientation, the worst-case bias combination considering bias alone may be different from the worst-case combination of bias and noise used for HPL<sub>c</sub>. Indeed, the greater the difference in orientation between the bias cloud and noise error ellipse, the greater benefit will be realized from a full search in a reduction of RPL<sub>c</sub>. 
After the worst case set of conspiring biases is obtained by search, the noise ellipse is added to this plot, as shown in Figure 16. Results appear in Table 7 below. The same bias combination was selected by both methods, so RPL\textsubscript{c} values are the same for both.

### Table 7: Simulated performance of RPL\textsubscript{c} methods

<table>
<thead>
<tr>
<th>RPL\textsubscript{c} Method</th>
<th>Value in m (99.99%)</th>
<th>Percent of simulated errors enclosed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact</td>
<td>4.732</td>
<td>99.989%</td>
</tr>
<tr>
<td>Approximation 1</td>
<td>4.732</td>
<td>99.989%</td>
</tr>
</tbody>
</table>
Summary of simulation results

The quantity $RPL_c (99.99\%)$ is 4.732 m, which is slightly larger than the $HPL_c (99.99\%)$ value 4.683 m, as expected. The difference between these measures increases for smaller values of enclosed probability $P_{pl}$, and decreases for larger $P_{pl}$ values. For example, $RPL_c (95\%)$ is 2.592 m while $HPL_c (95\%)$ is 2.500 m. At probability levels of 99.999% and higher, this 100,000-point simulation could not reliably provide results.

The approximations for $VPL_c$ and $HPL_c$ are observed to be overbounds, also as expected. At the 99.99% probability level for this simulation, the use of the approximations added about a half order of magnitude (i.e., 0.005%) of excess
overbounding error. Depending on the application and specified probability level, these approximations will continue to prove quite useful. The percentage of points enclosed in the protection level for a given simulation run follows a sampling distribution whose mean was not verified, but which appears to be the nominal value of the PL. This is the case for all such Monte-Carlo error simulations, regardless of the PL type.

The simulation helps to illustrate the differences between the various methods and to verify the algorithms. However, for the very high probability levels used in LAAS and similar applications, it cannot yet support meaningful conclusions. For that degree of analysis, real flight test data is essential. In the following section, the method for applying these protection levels to recorded data from Ohio University flight tests is described.

**Protection Level Parameters for Flight Test Data Analysis**

*Baseline for comparison: LAAS protection level parameters*

Values are selected for the LAAS performance baseline protection level computation from various parameters listed in [1]. Selections are made to match the DFD1 configuration as closely as possible. The ground accuracy designator “A1” curve is appropriate for a single non-LAAS survey grade GPS antenna. The airborne accuracy designator “B” curve is selected for noise. The “A” curve for airborne multipath is selected, in part because the “B” curve is still under development. The residual ionospheric delay term $\sigma_{\text{iono}}$ is set to 0.02 m, appropriate for an ionospheric spatial gradient that could occur over a 100-m distance from runway threshold to reference station. This is coincidentally equal to the contribution of higher order ionospheric delay terms remaining in a dual-frequency ionosphere-free implementation in which
ionospheric delay is removed from pseudorange and accumulated Doppler measurements. As stated previously, the DFD1 architecture is ionosphere-free only when the distance between the aircraft and reference station is zero, and only then because of differential processing. The term $\sigma_{\text{tropo}}$ is set to zero in post-processing because no severe weather fronts were observed during the flight test interval, and because the differential GPS architecture corrects for tropospheric error as a result of spatial decorrelation, especially if the aircraft accuracy is evaluated close to the reference station antennas. These parameter values are not intended to indicate a “best case” LAAS baseline, but a reasonable LAAS configuration as similar as possible to the DFD1 architecture proposed here.

**Protection level parameters used for proposed system under test**

In the proposed DFD1 differential GPS processing architecture, the ionospheric error is set to 0.02 m and the residual tropospheric error is set to zero, both as described above for the LAAS baseline. Since the code noise and multipath are exchanged for biases through the CNMP algorithm, however, the $\sigma_{\text{pr, gnd}, i}$ term cannot be reused as defined for LAAS. Instead, a different function to determine $\sigma_{\text{pr, gnd}, i}$, one that incorporates only thermal noise, is obtained from [105]:

$$\eta_{\text{PR}} = \Delta \delta \sqrt{\frac{BW}{\left(\frac{C/N_0}{10}\right)}}$$  \hspace{1cm} (46)

where

$$\Delta = \text{the GPS Coarse / Acquisition (C/A) code chip width (m)}$$
\[ \delta = \text{the correlator spacing (unitless)} \]
\[ BW = \text{the code tracking loop bandwidth (Hz)} \]
\[ C/N_0 = \text{the carrier-to-noise ratio (dB-Hz)} \]

For C/A tracking on narrow-correlator NovAtel OEM-4 receivers, with internal CSMOOTH value set to 20 (equivalent to a time constant of \( \tau = 3 \text{ s} \)), \( \Delta = 293 \text{ m} \), \( \delta = 0.1 \) and \( BW = 1/6 \text{ Hz} \). Because the thermal noise is further reduced by extended averaging in the CNMP algorithm, the equation above is divided by the factor \( \sqrt{T_{\text{avg}} / \tau_c} \), where \( T_{\text{avg}} \) is the length of the averaging time and \( \tau_c \) is the time constant for internal receiver smoothing, both in seconds, as given above. This yields the following form for the standard deviation of averaged thermal code noise:

\[
\eta_{PR} = \Delta \delta \sqrt{\frac{BW \times \tau_c}{2 \times T_{\text{avg}} \times 10^{\frac{C/N_0}{10}}}}
\]  

(47)

The term \( \sigma_{air,i} \) is determined in the same way.

Simulations of data smoothing using the factor \( \sqrt{T_{\text{avg}} / \tau_c} \) were performed with acceptable results. However, a more in-depth examination of smoothing effects, as in [106], is appropriate before this algorithm is implemented operationally.

**Method of Combining Conspiring Biases in Reference Station and Aircraft**

At both the reference station and the aircraft, range domain biases are combined in the worst possible way when transformed into the position domain. This is based on the general principle that probabilistic measures for safety of life cannot be diluted or
averaged across an ensemble. For this reason, the addition of biases is a worst-case vector sum rather than the root-sum-square combination used for noise.

The one exception to this principle in this analysis of flight data is that the range domain bias vectors from the reference station and the aircraft are in fact combined in a root-sum-square sense. For the symmetric bias combinations analyzed here, such that two bias combinations in each epoch have the same magnitude, the probability of the worst-case total bias combination being realized at a single location, either reference station or aircraft, is \(1 / 2^{(M-1)}\), where \(M\) is the number of satellites in the least-squares solution. As will be explained in the next chapter, the two largest bias components remaining after CNMP processing are those due to multipath and antenna group delay, and they are of the same order of magnitude. They are also independent. The probability of both of these occurring at once at a single location, either reference station or aircraft, therefore, is \([1 / 2^{(M-1)}]^2\), or \(1 / 2^{(2M-2)}\). For \(M = 6\), this probability is \(1 / 1024\). For \(M = 8\), it is \(1 / 16384\), and for \(M = 10\), it is \(1 / 262,144\). Although a rigorously performed combination of probabilities for the simultaneous worst-case noise (\(1 / 500,000,000\) separately for reference station and aircraft) and worst-case bias is not provided here, one can see that the joint probability of worst-case bias and noise being realized simultaneously on both reference station and aircraft is vanishingly small.

For this reason, reference station and aircraft bias bounds are assumed to be uncorrelated, which also implies independence for these normally distributed random variables. Thus, the root-sum-square method is used to combine them. The degree to
which reference station and aircraft biases are dependent, and the probability of their simultaneous worst-case combinations with the worst-case noise, is left to future work.
CHAPTER 5: MODIFIED CODE NOISE AND MULTIPATH ALGORITHM

As has been stated, the CNMP algorithm of [9-10] is modified in this chapter to calculate bias bounds based on real-time observations rather than preset values. This is also the first known use of the CNMP algorithm for both reference station and airborne system. The modified CNMP algorithm builds on the dual-frequency divergence-free approach to ionospheric error removal articulated by Euler and Goad in [107] and adapted to WAAS reference stations by Kee et al. in [108]. This algorithm’s derivation follows.

Derivation of Pseudorange Value Corrected for Code Noise and Multipath

Using the CMC observable, the modified CNMP algorithm removes the code noise and multipath errors from the pseudorange (i.e., code) measurements. As in [109], the pseudorange observation equation for a given satellite at time \( t \) is

\[
PR_{L_1}(t) = R(t) + I_{L_1}(t) + T(t) + \varepsilon_p(t) + \tau_{PR,L_1}(\theta(t), \psi(t)) + c\Delta t_{PR,L_1}(t) + \eta_{PR,L_1}(t)
\]

(48)

where

- \( R \) = the true geometric range from user to satellite (m)
- \( I \) = the ionospheric delay at the specified frequency (m)
- \( T \) = the tropospheric delay (m)
- \( \varepsilon \) = the satellite orbit error projection onto line of sight (m)
- \( \tau \) = the antenna delay for specified measurement and frequency (m)
- \( \theta \) = the satellite elevation angle (rad)
- \( \psi \) = the satellite azimuth angle (rad)
- \( c \) = the speed of light defined for GPS (m/s)
\[ \Delta t = \text{the receiver clock offset for specified measurement and frequency (sec)} \]

\[ \eta = \text{the noise, multipath and unmodeled errors for specified measurement and frequency (m)} \]

and with subscripts

- \( PR \) indicating pseudorange observable
- \( P \) indicating geometric projection onto user-to-satellite line of sight
- \( L \) = the specified GPS frequency, only L1 in this equation

The accumulated Doppler (AD) observation equation, which indicates time-integration of Doppler frequency shift measured since the satellite was first tracked by the receiver, is given by

\[
AD_{L1}(t) = R(t) - I_{L1}(t) + \frac{\lambda_{L1}}{2\pi} \psi(t) + T(t) + e_p(t) + \tau_{AD,L1}(\theta(t), \psi(t)) + N_{L1}(t)\lambda_{L1} + c\Delta t_{AD,L1}(t) + \eta_{AD,L1}(t) \tag{49}
\]

where

- \( N \) = the integer ambiguity number comprised of an unknown constant number of wavelengths between the satellite and user, plus a time-dependent number of wavelengths due to phase wrap-up roll-over
- \( \lambda_{L1} \) = the L1 wavelength (m)

and subscript

- \( AD \) indicates accumulated Doppler (also known as deltarange)

The CMC observable for the L1 frequency at time \( t \) is then formed by subtracting the AD frequency shift from the pseudorange:
\[ CMC_{L1}(t) = PR_{L1}(t) - AD_{L1}(t) \]
\[ = 2I_{L1}(t) - \frac{\lambda_{L1}}{2\pi} \psi(t) + \tau_{PR, L1}(\theta(t), \psi(t)) - \tau_{AD, L1}(\theta(t), \psi(t)) + \]
\[ + c(\Delta t_{PR, L1}(t) - \Delta t_{AD, L1}(t)) - N_{L1}(t)\lambda_{L1} + \eta_{PR, L1}(t) - \eta_{AD, L1}(t) \]  

Time-varying ionospheric delay—that is, ionospheric delay that has occurred since algorithm initialization when unknown but constant integer ambiguities \( N(t) \) are first used—is removed from the CMC using a combination of L1 and L2 AD measurements. This quantity is also termed differential delay by some authors, and describes a difference in measurements taken at different GPS epochs \[22\]. The L1 ionospheric delay is first written as

\[ I_{L1}(t) = \frac{f_{L2}^2}{f_{L1}^2 - f_{L2}^2} \left( I_{L2}(t) - I_{L1}(t) \right) \]  

where \( f_L \) indicates the GPS frequency.

The value of CMC corrected for the change in the L1 ionospheric delay that has occurred over time is obtained in three steps. First, the L1 and L2 AD measurements are differenced to produce an estimate for the expression \( I_{L2}(t) - I_{L1}(t) \):

\[ AD_{L1}(t) - AD_{L2}(t) = I_{L2}(t) - I_{L1}(t) + \frac{\lambda_{L1} - \lambda_{L2}}{2\pi} \psi(t) + \tau_{AD, L1}(\theta(t), \psi(t)) \]
\[ - \tau_{AD, L2}(\theta(t), \psi(t)) + c(\Delta t_{AD, L1}(t) - \Delta t_{AD, L2}(t)) \]
\[ + N_{L1}(t)\lambda_{L1} - N_{L2}(t)\lambda_{L2} + \eta_{AD, L1}(t) - \eta_{AD, L2}(t) \]  

Next, the right half of Equation (5) is substituted for \( I_{L2}(t) - I_{L1}(t) \) in Equation (51) to produce an expression for \( I_{L1}(t) \). The CMC L1 observable from Equation (50) is then corrected for changes over time in ionospheric delay by subtracting the term \( 2I_{L1}(t) \). The following expression results:
where

\[
\eta = \frac{2f_{L1}^2}{f_{L1}^2 - f_{L2}^2} \approx 3.0915
\]

This operation removes the L1 and L2 ionosphere terms, at the cost of adding the associated error and ambiguity terms introduced in Equation (52). The ionospheric-corrected CMC can now be rearranged and approximated as follows:

\[
CMC_{L1, corr}(t) \approx \eta_{PR,L1}(t) + \eta_{PR,L2}(t)\lambda_{L1}(t) + 4.09\eta_{AD,L1}(t) + 3.09\eta_{AD,L2}(t) - 4.09\tau_{AD,L1}(t) - 4.09\tau_{AD,L2}(t) + c\Delta t_{PR,L1}(t) - c\Delta t_{PR,L2}(t)
\]

This equation can be simplified using three assumptions. An antenna is selected whose variations in phase delay with azimuth and elevation, at frequencies L1 and L2, are negligible or can be calibrated. Receiver design can also ensure that clock offset for AD between L1 and L2, as well as clock offset between L1 pseudorange and L1 AD, is constant and therefore persists as a bias. Further, phase wrap-up corrections are performed for the time-varying difference between user heading and satellite azimuth angle that has occurred since algorithm initialization. The simplified form is

\[
CMC_{L1, corr}(t) \approx \eta_{PR,L1}(t) + \eta_{PR,L2}(t)\lambda_{L1}(t) + 4.09\eta_{AD,L1}(t) + 3.09\eta_{AD,L2}(t) + B
\]
The bias term $B$, which represents the remaining unresolved carrier phase ambiguity terms and clock offsets in Equation (55), has an unknown but constant value as long as continuous carrier tracking is maintained. The essence of the modified CNMP algorithm is that pseudorange noise and multipath become directly observable in the ionosphere-corrected CMC to the degree that this bias term can be accurately estimated. In this algorithm, the bias is estimated as the time average over $n$ seconds of the ionosphere-corrected CMC. Using $\Delta T$ as the measurement update rate in seconds, the initial and subsequent values of estimated bias are obtained as follows:

$$\hat{B}(0) = \text{CMC}_{L1,\text{corr}}(0)$$

$$\hat{B}(n\Delta T) = \frac{1}{n + 1} \sum_{i=0}^{n} \text{CMC}_{L1,\text{corr}}(i\Delta T); \ n > 0$$

(57)

Code noise and multipath can now be removed from the pseudorange. This is accomplished by subtracting Equation (56) from the pseudorange expression given in Equation (48), and then adding back in the bias estimate from Equation (57):

$$PR_{L1,\text{corr}}(t) = PR_{L1}(t) - \text{CMC}_{L1,\text{corr}}(t) + \hat{B}(t)$$

(58)

The resulting corrected pseudorange is

$$PR_{L1,\text{corr}}(t) = R(t) + I_{L1}(t) + T(t) + \varepsilon_P(t) + c\Delta t_{PR,L1}(t) + 4.09\eta_{AD,L1}(t) - 3.09\eta_{AD,L2}(t) - B + \hat{B}(t)$$

(59)

An examination of the pseudorange of Equation (48) and the corrected pseudorange of Equation (59) reveals that pseudorange noise, multipath, and antenna group delay are replaced by, or exchanged for, amplified carrier phase noise at L1 and L2 and bias estimation error $\hat{B}(t) - B$ that decreases, along with its symmetric bounds, over time. The standard deviation of the L1/L2 AD noise term, translated into the pseudorange
through subtraction of the CMC observable, is obtained through a root-sum-square
calculation as \( \sqrt{(4.09\sigma_{AD,1})^2 + (3.09\sigma_{AD,2})^2} \). For a typical 5-mm value of standard
deviation for AD noise on both frequencies, this term yields a standard deviation of 2.6

cm for “post bias exchange” pseudorange noise.

Having obtained an exceedingly small value of noise, the next step is to quantify
and remove the bias, which now contains the error, by adding an increasingly accurate
bias estimator value. If the pseudorange multipath and the antenna group delay errors are
zero mean over time, then the bias estimation error term converges to zero. The bound on
the bias will tend to decrease as a function of time as well. However, the bias bound
methodology does not depend on an assumption that the multipath is zero mean, as is
detailed below.

**Bound on Pseudorange Bias as a Function of Time**

The net bias term in the corrected pseudorange can be expressed as

\[
\hat{B}(n\Delta T) - B = \frac{1}{n+1} \sum_{i=0}^{n} \{ \eta_{PR,L1} (i\Delta T) + \tau_{PR,L1} (\theta(i\Delta T), \psi(i\Delta T)) + 3.09\eta_{AD,L2} (i\Delta T) - 4.09\eta_{AD,L1} (i\Delta T) \}
\]

(60)

This expression is initialized as follows:

\[
\hat{B}(0) - B = \eta_{PR,L1} (0) + \tau_{PR,L1} (\theta(0), \psi(0)) + 3.09\eta_{AD,L2} (0) - 4.09\eta_{AD,L1} (0)
\]

(61)

When initialized, the bias uncertainty is the combination of pseudorange noise and L1/L2
AD noise, to which the sum of the pseudorange multipath and the antenna group delay
error at initialization time is also added. Accordingly, an overbound of the bias term can
be determined from the worst-case combination of the realizations of the following items:
1. Pseudorange multipath
2. Antenna group delay
3. Pseudorange noise
4. AD L1 noise
5. AD L2 noise

Items 3, 4 and 5 can be combined in root-sum-square fashion into a single Gaussian distribution, which can then be bounded based on the overbound probability. These are considered term by term, followed by group delay and multipath.

**Bound on thermal noise component of initial bias**

As with the protection level computations in the previous chapter, the pseudorange thermal noise is expressed for a single GPS epoch as follows:

\[ \eta_{PR,L1} = \Delta \delta \sqrt{\frac{BW}{C/N_0}} \times 10^{\frac{1}{2}} \]

(62)

With typical values of \( C/N_0 = 45 \) dB-Hz, \( BW = 1/6 \) Hz and a correlator spacing of \( \delta = 0.1 \), the standard deviation of thermal noise is 4.7 cm. If the AD noise values for L1 and L2 are both conservatively estimated at 5 mm as before, the AD-derived noise terms in the initial pseudorange bias value combine to \( \sqrt{(2.6)^2 + (4.7)^2} = 5.2 \) cm. If a sigma multiplier of 6 is used, corresponding to a probability \( 2 \times 10^{-9} \) that the PL value will be exceeded, the bound for this term is 31.2 cm. Implicit in the tracking loop bandwidth \( BW = 1/6 \) Hz is that every third sample is uncorrelated. Therefore, the noise bound will decrease by a factor of \( \sqrt{T_{avg}/\tau_c} = 10 \) after averaging 100 uncorrelated updates, or \( T_{avg} = \)
300 s with a $\tau_c$ of 3 s. Thus, after five minutes, a combined noise term bound of 3.1 cm is achieved.

**Bound on antenna group delay component of initial bias**

A bound on the antenna group delay, listed as item 2 above, is derived from careful antenna design and characterization. A typical maximum value for group delay variations over the hemispherical pattern of a well-designed reference station antenna is 10 cm, and 20 cm for an airborne antenna. New antenna models may achieve lower bias floor values than these.

**Bound on multipath component of initial bias**

Finally, the pseudorange multipath error must be bounded. For the WAAS reference stations, a conservative model based on a priori observed data is used [9]. The approach taken below is different, in that it is based on observed variations in the ionosphere-corrected CMC, combined with siting considerations and a corresponding worst-case multipath threat model. For this reason, the CNMP formulation defined here is termed the modified CNMP algorithm.

**Derivation of Multipath Frequency**

If the ground antenna is sited closely above a flat ground area and is not near to above-ground obstacles, then the lowest multipath frequency results from ground multipath. The ground multipath amplitude can be bounded according to the following equation from [110]:

$$ MP_{PR} = \frac{\gamma}{(1-\gamma)} 2h \sin(\theta) \text{ m} $$

(63)
where

\[ \gamma = \text{multipath-to-direct signal strength ratio (assumed to be less than 1)} \]

\[ h = \text{the height of the GPS antenna above the ground surface (m)} \]

It is now possible to calculate the largest multipath error possible at the antenna by using the antenna multipath-to-direct ratio from the antenna pattern, and the height of the antenna above the ground. This was implemented for the NovAtel PinWheel 600 antenna at a height of 3 m above the ground. The results are shown in Figure 17, in which it may be observed that ground multipath error in this installation will never exceed 2 m assuming an elevation mask angle of five degrees.
At initialization, then, the largest possible bias error in the pseudorange measurements is obtained for the flight test data by combining the multipath, group delay and noise bounds. Using values derived above, this initial maximum bias bound for the aircraft is $2 + 0.2 + 0.31 = 2.51$ m. For the reference station, it is 2.41 m due to the smaller antenna group delay bias.

*Decreasing bound on multipath bias as a function of time*

The multipath component of this bias error bound can be reduced over time by examining variations of the ionosphere-corrected CMC. This is possible because multipath from moving satellites is not constant. This bias reduction is obtained by first
calculating the multipath fading frequency, which is obtained by differentiating the path
delay, $2h \sin(\theta)$, with respect to time and converting units to wavelengths:

$$f_{\text{fading}} = \frac{2h \dot{\theta}}{\lambda} \cos(\theta) \text{ Hz} \quad (64)$$

Before using a satellite, the algorithm verifies that the fading time period is less than
1000 s. Most satellites meet this requirement, particularly those acquired just as they rise
above the mask angle.

Antenna group delay is not included in this algorithm, but rather is treated as
constant during the 1000-s evaluation interval. Because a satellite typically requires three
to six hours to transit completely through the antenna’s fixed hemispherical pattern, time
variation in antenna group delay bias has a much slower frequency than that for
multipath.

The multipath bound is found by evaluating the ionosphere-corrected CMC over
the maximum fading time period of 1000 s. To reduce high-frequency multipath
components, a low-pass filter with bandwidth of $1/1000$ Hz is applied to $CMC_{L1, \text{corr}}$, and
the output amplitude is amplified to compensate for filter attenuation at the cut-off
frequency. The maximum and minimum values of the compensated filter
output $CMC_{L1, \text{ corr, filt}}$ are determined and the difference is assigned as the conservative
multipath bound:

$$M_{P_{\text{bound}}} = \max \left\{ CMC_{L1, \text{ corr, filt}} \right\} - \min \left\{ CMC_{L1, \text{ corr, filt}} \right\} \quad (65)$$
where the max{} and min{} operators determine the maximum and minimum values over the 1000-s time interval, respectively. During the first 1000 s, the worst case multipath value is used and the bias value is estimated using all the data available at each iteration. When the 1000-s initialization time has completed, the algorithm continues to look for a smaller multipath bound over successive (i.e., sliding) 1000-s time intervals. If a smaller bound is discovered, that value is recorded as the bound and the filter uses it for future comparisons. Simultaneously, the previous estimate of bias value is also replaced with the new estimate. Any larger peak-to-peak values of multipath subsequently observed are interpreted as not being part of the bias (i.e., if it were part of the bias, it wouldn’t be changing) and are discarded, and the estimate of bias value is not updated. No zero mean multipath assumption is made in computing the bias bound; only the peak-to-peak difference is used.

The 1000-s time interval may possibly have a significant negative effect on system continuity. However, if the CNMP algorithm is started well prior to the beginning of the precision approach, and if continuous carrier phase tracking is maintained for at least four satellites (or if the signal is patched using inertial data), then cycle slips or other tracking difficulties result in a relatively graceful degradation both to accuracy and to PL values. This is because an abrupt return to the initial bias values of 2.51 m or 2.41 m for a particular satellite is accompanied by a simultaneous de-weighting of that satellite in both position and PL solutions.

An example of the modified CNMP bounding algorithm’s performance is shown in Figure 18. Ohio University flight test data at the reference station from SV 28 in a
2000-s window is shown. At initialization, the satellite’s elevation is 15 degrees (rising), its azimuth is 271 degrees with respect to true North and its $C/N_0$ value is 40 dB-Hz.

The upper plot shows the CMC, corrected for ionospheric delay only, with code noise and multipath present (noisy plot) and removed (smoothed plot). Since the bias of these two plots is constant as long as carrier phase tracking is continuous, but is unknown, these values are plotted relative to the CMC value obtained upon algorithm initialization. At iteration 629, a lock reset is reported by the GPS receiver, perhaps caused by an overflight of an aircraft over the reference station. Upon reestablishing lock at the next second, the initial bias in the receiver data is still unknown, but obviously has changed. The CNMP algorithm is restarted and smoothes data throughout the rest of the 2000-s period shown.

In the lower plot, the maximum bias bound is initialized at the reference station maximum of 2.41 m. After the successful 1000-s period of continuous carrier phase tracking has elapsed, as described above, the algorithm first reports a bound of 0.29 m at iteration 1632. At the next second, it reports a bound of 0.18 m. By iteration 1650, the bound has reached the floor value of 0.10 m and remains there for the rest of the 2000-s period.
In summary, the modified CNMP algorithm is initialized for a worst-case bias of 2.51 m (2.41 m for the reference station). After 1000 s, a more accurate bias bound is calculated using Equation (65). During this interval, the antenna group delay bias component is kept constant, and the bias component that originated as noise decreases as a function of the number of independent measurements. The only remaining noise in the corrected pseudorange that has not been exchanged for a bias is the contribution from L1/L2 AD noise, as given in Equation (59). Table 8 summarizes values for key parameters used in the modified CNMP algorithm.
Table 8: Modified CNMP algorithm parameter values for flight test data

<table>
<thead>
<tr>
<th>Bias Parameter</th>
<th>Values</th>
<th>Bound at Initialization (better than 2×10⁻⁹)</th>
<th>Bound after Initialization (better than 2×10⁻⁹)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pseudorange multipath</td>
<td>Figure 17</td>
<td>2.0 m</td>
<td>Eqn (65), ≤ 2.0 m</td>
</tr>
<tr>
<td>Antenna group delay</td>
<td>–0.1 to 0.1 m (ground)</td>
<td>0.1 m (ground)</td>
<td>0.1 m (ground)</td>
</tr>
<tr>
<td></td>
<td>–0.2 to 0.2 m (air)</td>
<td>0.2 m (air)</td>
<td>0.2 m (air)</td>
</tr>
<tr>
<td>Pseudorange noise (from $\hat{B}$)</td>
<td>Eqn (62)</td>
<td>$6 \times$ Eqn (62)</td>
<td>--</td>
</tr>
<tr>
<td>AD L1 noise (from $\hat{B}$)</td>
<td>$4.09 \times 0.5 \text{ cm (} \sigma \text{)}$</td>
<td>$6 \times 4.09 \times 0.5 \text{ cm}$</td>
<td>--</td>
</tr>
<tr>
<td>AD L2 noise (from $\hat{B}$)</td>
<td>$3.09 \times 0.5 \text{ cm (} \sigma \text{)}$</td>
<td>$6 \times 3.09 \times 0.5 \text{ cm}$</td>
<td>--</td>
</tr>
<tr>
<td>Noise combined (for 45 dB-Hz) from $\hat{B}$</td>
<td>5.2 cm (σ)</td>
<td>31.2 cm</td>
<td>$n_{tot} = 31.2 \text{ cm} / \sqrt{n/3}$</td>
</tr>
<tr>
<td>Corrected pseudorange bias bound</td>
<td>--</td>
<td>2.41 m (ground)</td>
<td>Eqn (65) + 0.1 + $n_{tot}$ (air)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.51 m (air)</td>
<td>Eqn (65) + 0.2 + $n_{tot}$ (ground)</td>
</tr>
<tr>
<td>Corrected pseudorange noise sigma</td>
<td>Eqn (59) and following</td>
<td>2.6 cm (σ)</td>
<td>2.6 cm (σ)</td>
</tr>
</tbody>
</table>

The corrected pseudorange noise sigma and the corrected pseudorange bias bound are used in the protection level equations. In processing this flight test data, the aircraft implementation is also based on the conservative ground station assumptions for multipath. The airborne contribution could be further reduced by taking its higher multipath fading frequencies and smaller reflective surfaces into account.

In addition to the carrier phase lock and cycle slip detectors provided by the GPS receivers, a basic cycle slip detection algorithm is implemented in software prior to execution of the CNMP algorithm. Near-simultaneous cycle slips on multiple satellites
could likely have a negative impact on system continuity, but such a condition was not observed in the flight test results presented below. Before actual implementation, a more detailed analysis of this and other integrity components of the DFD1 architecture is needed.
CHAPTER 6: CARRIER PHASE POSITION DOMAIN SMOOTHING ALGORITHM

Origin and Benefits

Instead of using code-carrier techniques specified for ground and air LAAS processing in [1], smoothing in the DFD1 processing architecture is accomplished solely in the aircraft. A position domain technique first described in [11] is used. Unsmoothed pseudorange corrections (i.e., not smoothed using carrier phase data) and AD data from the reference receiver, along with unsmoothed pseudoranges and AD data from the airborne receiver, are combined to propagate the aircraft’s position from the previous time to the current time with centimeter-level accuracy.

This position domain smoothing approach offers several important benefits. First and perhaps most importantly, the ionospheric divergence associated with range domain smoothing in the presence of ionospheric spatial gradients in differential systems is avoided because all processing is performed in the aircraft, and because the smoothing algorithm has no memory beyond the previous GPS epoch.

A second benefit is that a virtual “rail in the sky” is created to define the aircraft approach path. The smoothness of the rail is provided by the centimeter-level accuracy of the position domain updates, which is in turn made possible by the use of carrier phase data. The position solution is thus kept free from abrupt changes that would otherwise occur when satellites enter or leave the visible constellation. This is especially important for an aircraft autoland computer, which might execute a dangerous nose-down command
on final approach in response to such a step change in GPS position data. The rail is kept centered in its true position by continual updates from the code phase solution.

Integrity of the position solution is enhanced by the carrier phase lock and cycle slip detectors provided by the GPS receivers. The cycle slip detection algorithm implemented in software prior to the CNMP algorithm adds to this integrity. Before actual implementation, a more detailed analysis of these and other integrity components of the DFD1 architecture is needed.

Two additional advantages serve to enhance system continuity. As soon as a satellite’s carrier phase data is available, it may be used to propagate the differentially-corrected aircraft position solution. Transition effects remain a few centimeters or less for a typical constellation. Furthermore, the centimeter-level accuracy of the relative position solution from one measurement epoch to the next enables the patching of pseudoranges following a temporary loss of lock. The specification for Category I non-federal LAAS ground facilities already allows such patching to accommodate aircraft overflights by specifying an update rate of 0.5 s, not to exceed 1.0 s (see paragraph 3.2.1.2.8.5.1 of [111]). Previous specification versions described the patching process in more detail, but the most recent document gives more leeway to the manufacturer. The method presented here does, however, require continuous carrier phase tracking of at least four satellites to avoid an algorithm restart. An inertial sensor could be used to patch a temporary loss of carrier phase data.
The position domain algorithm was previously summarized and tested in [11], but no equations were included. They are provided below, using Huang’s setup and notation from [112].

**Algorithm Description**

Let the single difference between simultaneous carrier phase measurements of satellite $i$ by users at points A and B be defined as follows:

\[
SD_{AB,i}(t) = R_{AB,i}(t) + c\Delta t_{AD,AB,L}(t) + N_{AB,i}\lambda_L + \varepsilon_{P,AB,i}(t) - I_{AB,i}(t) + T_{AB,i} + mp_{AD,AB,i}(t) + \eta_{AD,AB,i}(t)
\]

where

- $R$ = the true geometric range from user to satellite (m)
- $c$ = the speed of light defined for GPS (m/s)
- $\Delta t$ = the receiver clock offset between identical measurements on the two receivers at points A and B (s)
- $N$ = the integer difference in the number of wavelengths between the satellite and users at points A and B
- $\lambda$ = the wavelength of the specified GPS frequency (Hz)
- $\varepsilon$ = the difference in satellite orbit error projections onto line of sight (m)
- $I$ = the difference in ionospheric delay (m, with – sign indicating advance)
- $T$ = the difference in tropospheric delay (m)
- $mp$ = the difference in multipath error (m)
- $\eta$ = the difference in noise error (m)

and with subscripts
\[ AB \quad \text{indicating point B relative to point A} \]
\[ AD \quad \text{indicating accumulated Doppler (also known as deltarange)} \]
\[ P \quad \text{indicating geometric projection onto user-to-satellite line of sight} \]
\[ L = \text{the GPS frequency, } L_1 \text{ only for the single differences computed here} \]
\[ i = \text{the index of usable satellites (1, 2, \ldots, } n) \]

It is assumed that known errors from sources such as antenna phase differences, which depend on azimuth and elevation, and phase wrap-up have been removed. Given the short baseline assumption, the terms \( I, \varepsilon, \) are each on the order of one or two centimeters or less and may be excluded from consideration in these position calculations. The same is true for the \( mp \) term, which is difficult to model in any event. A small residual ionospheric error sigma of 0.02 m is accounted for, however, in the protection level computations. Differential GPS processing accounts for differential height decorrelation between receivers and between the current and previous epoch. Since this error reduction is accomplished in previous processing stages using dual-frequency data, all position domain smoothing may be accomplished solely using \( L_1 \). Satellite inter-frequency and inter-code biases are also assumed to cancel in the differential corrections. Therefore, the single difference for satellite \( i \) now becomes

\[
SD_{AB, L_1, i} (t) = R_{AB, i} (t) + c\Delta t_{AD, AB, L_1} (t) + N_{AB, L_1, i} A_{L} + \eta_{AD, AB, L_1, i} (t) + T_{AB, L_1, i} (t) \tag{67}
\]

A geometric illustration of the single difference is appropriate at this point. Let the two GPS receivers located at points A and B be separated at time \( t \) by baseline vector \( \mathbf{\bar{b}}_{AB} (t) \), as shown in Figure 19 below. Let the line-of-sight vectors from A and B to
satellite \(i\), which is located at point \(S\), be \(\mathbf{e}_A(t)\) and \(\mathbf{e}_B(t)\), respectively. Then the difference in length between these two vectors, which is the difference in range from each user to the satellite, may be determined using the inner product:

\[
R_{AB,i}(t) = \mathbf{e}_{A,i}(t) \cdot \mathbf{b}_{AB}(t)
\]

(68)

In terms of line segments, range difference \(\overline{AO} = \overline{AS} - \overline{OS}\). It is apparent that \(\overline{OS} = \overline{BS}\).

**Figure 19**: Range difference \(\overline{AO}\) from users at points \(A\) and \(B\) to satellite at point \(S\) and at time \(t\), with parallel lines of sight

For mathematical convenience and as Figure 19 illustrates, use of the inner product includes the assumption that vectors \(\mathbf{e}_A(t)\) and \(\mathbf{e}_B(t)\) are parallel. In reality, vectors \(\mathbf{e}_A(t)\) and \(\mathbf{e}_B(t)\) are not parallel for non-zero baselines, so the range difference must be corrected:

\[
R_{AB,i}(t) = \mathbf{e}_{A,i}(t) \cdot \mathbf{b}_{AB}(t) - \text{corr}_{AB}(t)
\]

(69)
In the true geometry shown in Figure 20, line segment $\overline{AC}$ is the true range difference. Then segments $\overline{CS}$ and $\overline{BS}$ must be equal for $\overline{AC}$ to be their difference. The nonlinearity correction, $corr_{AB}(t)$, is represented by line segment $\overline{OC}$. This length must be subtracted from range difference $\overline{AO}$, which is calculated by the inner product, to avoid a systematic position error that increases with distance $\overline{AB}$. The correction will be derived shortly.

![Figure 20: True range difference AC from users at points A and B to satellite at point S, all at time t](image)

At algorithm initialization, the best relative position estimate available is used for mobile point B. This is the unsmoothed differential solution for the receiver at point B, by definition an estimate of column vector $\overline{b}_{AB}(t)$. A locally level, east-north-up
coordinate frame is used. Since point A is always the origin, this formulation is equally applicable to differential systems, where A is stationary, and relative systems, where A may move as well as B. True user-to-satellite line of sight column vectors \( \mathbf{e}_A(t) \) and \( \mathbf{e}_B(t) \) are now exchanged for estimated line of sight column vectors \( \mathbf{r}_A(t) \) and \( \mathbf{r}_B(t) \). The vector \( \mathbf{r}_B(t) \) is obtained using the estimated position of B for time \( t_1 \), which is the first GPS measurement epoch:

\[
\mathbf{r}_B(t_1) = \mathbf{r}_A(t_1) - \mathbf{b}_{AB}(t_1)
\]  

(70)

The nonlinearity correction is calculated for time \( t_1 \) as in [112], and is saved for further use:

\[
corr_{AB}(t_1) = \frac{\mathbf{r}_B^T(t_1) \mathbf{r}_A(t_1)}{||\mathbf{r}_A(t_1)||}
\]  

(71)

The computations above are repeated for each of \( n \) satellites, and the geometry matrix \( \mathbf{H} \) is constructed for point B at time \( t_1 \), with user-to-satellite unit vector elements in the first three columns of each row and the integer value 1 in column 4. The \( i \)-th row of \( \mathbf{H} \) at time \( t_1 \) is then as follows:

\[
\mathbf{H}_{i,..}(t_1) = \begin{bmatrix}
\frac{r_{B,i,\text{east}}(t_1)}{||\mathbf{r}_{B,i}(t_1)||} & \frac{r_{B,i,\text{north}}(t_1)}{||\mathbf{r}_{B,i}(t_1)||} & \frac{r_{B,i,\text{up}}(t_1)}{||\mathbf{r}_{B,i}(t_1)||} & 1
\end{bmatrix}
\]  

(72)

At time \( t_2 \), another set of GPS measurements is taken simultaneously by the two receivers at point A and B. In this case, however, position \( B(t_2) \) is assumed to be
different than $B(t_1)$, relative to point $A$. Additionally, the position $S(t_1)$ of the $i$-th satellite has changed to $S(t_2)$. Geometrically, this appears as shown in Figure 21 below.

Figure 21: Propagation of position $B$ from time $t_1$ to $t_2$

A new estimated relative position is obtained for $B(t_2)$, namely the code phase differential solution for $\mathbf{b}_{AB}(t_2)$. The quantities $\mathbf{r}_A(t_2)$, $\mathbf{r}_B(t_2)$, $corr_{AB}(t_2)$ and $H(t_2)$ are then computed as above. The position propagation vector $\Delta \mathbf{b}(t_2 - t_1)$, the desired quantity in this derivation, is obtained as follows.

The single difference defined above for the $i$-th satellite between points $A$ and $B$ is here restated in terms of the AD observables in units of meters, at times $t_1$ and $t_2$: 
A double difference needed to obtain the propagation vector $\Delta b(t_2 - t_1)$ is then formed by differencing these two. Substituting, this double difference is

$$DD_{AB,L1,i}(t_2 - t_1) = AD_{A,i}(t_2) - AD_{B,i}(t_2) - (AD_{A,i}(t_1) - AD_{B,i}(t_1))$$

(75)

For convenience in processing, the double difference may be rearranged as follows:

$$DD_{AB,L1,i}(t_2 - t_1) = AD_{A,i}(t_2) - AD_{A,i}(t_1) - (AD_{B,i}(t_2) - AD_{B,i}(t_1))$$

$$= SD_A(t_2 - t_1) - SD_B(t_2 - t_1)$$

(76)

The double difference is then corrected for tropospheric spatial decorrelation between reference and airborne receivers from time $t_1$ to time $t_2$:

$$DD_{AB,L1,i}(t_2 - t_1) = SD_A(t_2 - t_1) - SD_B(t_2 - t_1) - \left(\hat{T}_A(t_2 - t_1) - \hat{T}_B(t_2 - t_1)\right)$$

(77)

Similarly, the double difference is corrected to remove nonlinearities not accounted for in the single difference computations:

$$DD_{AB,L1,i}(t_2 - t_1) = SD_A(t_2 - t_1) - SD_B(t_2 - t_1) - \left(\hat{T}_A(t_2 - t_1) - \hat{T}_B(t_2 - t_1)\right)$$

$$- \left(\hat{T}_A(t_2) - \hat{T}_B(t_1)\right) - (corr_{AB}(t_2) - corr_{AB}(t_1))$$

(78)

Column vector $dd$ is formed with rows $i = (1, 2, ..., n)$, each of which is a corrected double difference observable $DD_i$. Since the calculation is understood as being from A to B and for frequency L1, the remaining subscripts are suppressed. The matrix of partial derivatives in the position domain is calculated using unweighted least squares:

$$G(t_2) = \left(\mathbf{H}^T(t_2)\mathbf{H}(t_2)\right)^{-1}\mathbf{H}^T(t_2)$$

(79)
An additional geometry correction is created using the best unsmoothed estimate of the position of the user at point B at time $t_1$:

$$ L(t_2 - t_1) = \begin{bmatrix} H(t_2) - H(t_1) & \hat{b}_{AB}(t_1) \\ 0 \end{bmatrix} $$

By applying this correction, the “snapshot” position propagation vector from time $t_1$ to $t_2$ for the user at point B is then obtained as follows:

$$ \Delta \hat{b}(t_2 - t_1) = G(t_2) (dd(t_2 - t_1) - L(t_2 - t_1)) $$

The scalar residual of this estimate is formed using the $(n \times n)$ identity matrix $I$ and the vector norm operation, as follows:

$$ q(t_2 - t_1) = \left| (I - H(t_2)G(t_2))(dd(t_2 - t_1) - L(t_2 - t_1)) \right| $$

If the residual $q$ is larger than the empirically derived threshold of 0.2 m and at least six satellites are in the solution, a RAIM operation is performed by excluding the satellites one at a time. If a satisfactory subset of satellites is found, the satellite with excessive error is removed and the position propagation vector $\Delta \hat{b}$ is recalculated. Then at time $t_2$, the propagated position is created:

$$ \hat{p}_{AB}(t_2) = \hat{b}_{AB}(t_1) + \Delta \hat{b}(t_2 - t_1) $$

Otherwise, no smoothed position is returned for time $t_2$, the snapshot differential solution $\hat{b}_{AB}(t_2)$ is used for propagated position $\hat{p}_{AB}(t_2)$ and the algorithm is restarted.

The significance of the expression $\Delta \hat{b}(t_2 - t_1)$ is that an instantaneous, very high-quality integrated velocity estimate is obtained for the user at point B over the time interval $t_1$ to $t_2$. The same type of geometry corrections that hold errors in average
velocity to less than 1 cm/s in [113] have been painstakingly made. Furthermore, this estimate is not derived from a filter, but it may be used as an input to other filters. For example, this estimate can be used to calibrate an inertial navigation system or smooth noisy position states. In the current derivation, it is used to smooth the differential position derived from pseudorange measurements at time $t_2$. The smoothed position is thus created using an alpha filter:

$$\hat{s}_{AB}(t_2) = \alpha \hat{b}_{AB}(t_2) + (1 - \alpha)\hat{p}_{AB}(t_2)$$

(84)

Here smoothing weight $\alpha = 0.005$ is used, corresponding to an empirically derived smoothing time constant of 200 s for the 1-s sample rate. The unsmoothed differential code phase solution, which is accurate but also contains step changes when satellites enter and leave the constellation, is smoothed by the low-noise carrier phase update. Put a different way, the longer-term error of the ultra low-noise carrier phase propagation method is kept very close to zero.

In the algorithm’s next iteration, the $t_2$ values just computed become $t_1$ values. The smoothed differential position $\hat{s}_{AB}(t_2)$ from the most recent iteration replaces code phase estimate $\hat{b}_{AB}(t_1)$ in the new iteration. This continues as long as continuous carrier phase tracking exists for at least four satellites. If this condition is not met, then the algorithm is restarted.

**Use of Unweighted and Weighted Least Squares for Post-CNMP Processing**

The unweighted least squares method is used for carrier phase propagation in the previous section. This is because the carrier phase multipath and thermal noise have
approximately the same small magnitude, and because a good model for weighting carrier phase multipath is unknown. Code phase measurements, however, are weighted by $1 / \mu_i$ because the post-CNMP bias is so much larger than the post-CNMP noise, as described in chapter 5. This has a significant effect in reducing position error. The floor values of $B_{\text{max},i}$ described in the previous section prevent excessively large weighting values from being included.

**Baseline for Comparison: LAAS 100-s Range Domain Smoothing**

The LAAS code phase observable in ground and air is smoothed using carrier phase data, as suggested in [1]. The equation is reproduced here for comparison. The smoothed pseudorange is

$$P_n = \alpha P + (1 - \alpha) \left( P_{n-1} + \frac{\lambda}{2\pi} (\phi_n - \phi_{n-1}) \right)$$

(85)

where

- $P_n$ = the smoothed pseudorange (m)
- $P_{n-1}$ = the previous smoothed pseudorange (m)
- $P$ = the raw pseudorange measurement (m, see [1] for more details)
- $\lambda$ = the L1 GPS wavelength (m)
- $\phi_n$ = the carrier phase (AD) (rad)
- $\phi_{n-1}$ = previous carrier phase (AD) (rad)
- $\alpha$ = the filter weighting function equal to sample interval divided by $\tau = 100$ s
This chapter has described the algorithm by which smoothed differential positions are obtained. Next, this algorithm is evaluated using recorded flight data. Before the results are presented, the configuration and procedures governing the flight test are first presented in the following chapter.
CHAPTER 7: FLIGHT TEST DESCRIPTION

General Description

Six 150-s straight-in approaches were flown at the Ohio University (OU) airport in Albany, Ohio in the late morning and early afternoon of October 29, 2008. The test period constituted 1 hour, 46 minutes and 18 seconds of data, or 6,378 consecutive GPS epochs. The weather was sunny and cool with a broken cloud layer at about 4,000 ft AGL, with winds from 10 to 20 kts. The test aircraft was the OU Avionics Engineering Center’s King Air C90SE, as shown in Figure 22 during a different test event.

Figure 22: Ohio University Avionics Engineering Center’s King Air C90SE test aircraft
Equipment Configuration

The airborne system included two NovAtel OEM-4 L1L2W GPS receivers connected to one dual frequency GPS patch antenna (the S67-1575-14 model by Sensor System Inc.), a laptop computer and a Harris VHF data broadcast (VDB) receiver connected to a wire antenna on the underside of the aircraft. The location of the GPS antenna on the aircraft is indicated in Figure 22. The computation engine was Matlab v2006b in Windows XP, on a 2.8 GHz Pentium 4 Mobile processor with 512 MB RAM. Real-time interfaces used custom Matlab code and serial-to-TCP/IP converters.

NovAtel binary GPS data from the airborne receivers, data received over the VDB and differential positions—smoothed and unsmoothed—computed aboard the aircraft were recorded every second. Of these, however, only the recorded raw GPS data is used in this analysis. No WAAS data is used. While data transmitted to the aircraft was unsmoothed as described here, the bias and revised sigma parameters needed to compute the composite protection levels were not transmitted in real-time. Rather, they were added in post-processing, along with the CNMP-processed pseudoranges. The results presented here are considerably better than those produced in real time by the October 2008 versions of this software.

The reference station consisted of four NovAtel OEM-4 L1L2 receivers connected to a single NovAtel Pinwheel 600 L1L2 antenna, a desktop computer running Matlab on Windows XP, and a Harris VDB transmitter broadcasting data at 114.450 MHz at 70W effective radiated power with an elliptically polarized dipole antenna. The same Matlab computational engine used in the airborne system was used in the reference
station, and six TCP/IP-to-serial converters were used in parallel in real time, controlled from Matlab. NovAtel binary data was recorded at 1 s intervals.

The Ohio University LAAS shelter, along with its mounted antennas, appears in Figure 23. The GPS antenna is 3 meters above ground level, at an ellipsoidal height of 199.448 m. The edge of the GPS shelter visible to the right, with the two GPS antenna mounting poles, is oriented parallel to the runway at 241 degrees. The shelter is about 200 m southeast of the runway touchdown point on level ground free of obstructions or other structures except for the four LAAS antenna mounts. In post-processing this flight data, data from only one reference receiver was used for the sake of simplicity. The data from this receiver was indistinguishable in terms of performance or desirability from the data of the other three. It is recognized that the 200 m distance between the touchdown point and reference station is larger than the 100 m limit imposed above for integrity purposes. However, since no unusual ionospheric activity was present during this proof-of-concept flight test, the extra 100 m is ignored.
The location of the GPS antenna in updated WGS-84 coordinates (ITRF00) is 39 degrees, 12 minutes and 48.03919 seconds N latitude; 82 degrees, 13 minutes and 16.65261 seconds W longitude; and ellipsoidal height 199.448 m.

**Flight Tracks and Satellite Geometries**

The ground track for the flight test period appears in Figure 24, and the flight profile for each of the six 150-s approaches appears in Figure 25. On each approach, the pilot flew a standard Instrument Landing System (ILS) approach path with a three-degree glideslope, using ILS guidance. Although DGPS data was present in real time in the aircraft, it was not presented to the pilot or used for guidance. Two approximately ten-minute periods, one just after approach #2 and the other just after approach #4, were spent on the ground taxiways for crew change-outs.
Figure 24: King Air ground track for October 29, 2008 flight
The number of satellites, as well as the vertical dilution of precision (VDOP) and horizontal dilution of precision (HDOP) satellite geometry indicators, appears in Figure 26 for the reference station. The airborne values are virtually the same.
**Truth Processing**

Airborne truth values were obtained using Waypoint Consulting’s GrafNav software, which provides a post-processing kinematic solution from recorded reference station and airborne data. All but 13 of the 6,378 GPS epochs in this data set yielded an ambiguity resolved solution. None of those 13 epochs is consecutive, and each provides a “next best” stable float solution. Advertised GrafNav accuracies for ambiguity resolved solutions in properly sited, dual-frequency, kinematic receiver installations similar to this one are $2 \text{ cm} + 1 \text{ mm/km}$ for baselines of 0 to 5 km, and $5 \text{ cm} + 2 \text{ mm/km}$ for baselines of 5 to 35 km.
Performance Standard

The performance standard used in this analysis is patterned after the Ground Based Augmentation Service Level D (GSL-D), as defined in [1]. One exception is made to provide a more generic analysis, and to account for the use of $M_m = 1$ reference antenna, which is not addressed in the standard. This exception is that the Horizontal Missed Detection Multiplier used here is 6.0, for $P_{HMI} < 2 \times 10^{-9}$, rather than the GSL-D Lateral Missed Detection Multiplier of 6.9, which corresponds to $P_{HMI} < 5 \times 10^{-12}$. The lateral error sigma (i.e., error aligned in the cross-track direction of a particular runway) is always less than or equal to the maximum horizontal error sigma, so the use of HPL rather than LPL is considered an overbound. The GSL-C and GSL-D standards are reproduced in Table 9 below for comparison. Continuity is not examined or included.

<table>
<thead>
<tr>
<th>GSL</th>
<th>Accuracy</th>
<th>Integrity</th>
<th>Lateral Missed Detection Multiplier</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lateral NSE Accuracy 95%</td>
<td>Vertical NSE Accuracy 95%</td>
<td>Integrity Probability</td>
</tr>
<tr>
<td>C</td>
<td>16.0 m</td>
<td>4.0 m</td>
<td>1 - $2 \times 10^{-7}$ in any 150 s</td>
</tr>
<tr>
<td>D</td>
<td>5.0 m</td>
<td>2.9 m</td>
<td>1 - $1 \times 10^{-9}$ in any 15 s vert, 30 s lat</td>
</tr>
</tbody>
</table>
CHAPTER 8: FLIGHT TEST RESULTS

Results for Entire Flight

Analysis of the flight data shows that the CNMP and CPDS algorithms yield significant reductions in all measures of error when compared to the baseline LAAS 100-s smoothing algorithm. Over the entire data set, 95% vertical error is reduced from 0.80 m to 0.32 m. Similarly, 95% horizontal error is reduced from 0.41 m to 0.33 m over the same period. Figure 27 and Figure 28 illustrate these error performance improvements.

The time intervals of the six approaches are superimposed over the data sets. As noted in the previous chapter, two approximately ten-minute crew change-out events on the runway ramp occurred immediately following approaches two and four. The proximity to the metal-roofed airport terminal building during these two intervals may be a cause of the relatively large horizontal errors in the LAAS dataset during these times, as the multipath assumptions for both architectures are probably not met. However, the data from these periods are retained in the overall flight statistics, and may be considered a preview of sorts for compliance with Category IIIc, which includes surface operations. Of course, they do not overlap with the precision approaches, and the approach statistics provide more definitive measures of approach performance than statistics for the whole flight. The small 2-5 cm jumps between adjacent GPS epochs evident in the CNMP errors in these plots are due to changes in the constellation, weights for bias bounds for individual satellites, and/or small jumps in the estimated bias value. Aircraft maneuvering or overflights of the reference station from nearby aircraft may have produced these.
Figure 27: Vertical accuracy and 95% error performance for LAAS (with 1 reference receiver) and DFD1 architectures, entire King Air flight test
LAAS PLs are also computed for comparison as per the LAAS MASPS [1] using one reference station antenna/receiver pair, with additional minor modifications: the noise sigma estimates used in the flight test for the CNMP-based DFD1 architecture (ground and air), which are derived from real-time carrier-to-noise ratio (CNR) values reported from the GPS receivers, are also used as the LAAS noise sigma estimates. This modified LAAS architecture is intended to enable a fair comparison with the modified CNMP-based architecture. In addition, the residual ionospheric and tropospheric error sigma values are set to zero in both sets of PLs computations, in keeping with the
assumption that they are removed in this architecture only in differential processing and are thus valid only in the immediate vicinity of the reference station.

The vertical and horizontal protection level (VPL and HPL) results are shown in Figure 29 and Figure 30 below. The jumps in the PLs are due to satellite constellation changes. A dramatic improvement is seen in the protection levels for the whole-flight dataset. During the first 1000-s initial averaging period of the CNMP algorithm, the CNMP protection levels exceed those derived from the LAAS 100-s smoothing algorithm. During this time, the CNMP algorithm defaults to the LAAS protection levels. Thus, CNMP mean protection level calculations made over the whole data set include an initial portion of PL data from LAAS. After this initial period, the CNMP protection levels are less than half of the size of the LAAS protection levels. Even when using the largest overbounding approximations, the mean vertical protection level is reduced from 5.90 m to 2.60 m, a reduction of 59.6%. Similarly, the mean horizontal protection level drops from 3.24 m to 1.38 m, or 57.4%. Figure 29 and Figure 30 demonstrate this improvement, with the wide green lines designating the most conservative approximation from which these values are taken. The abrupt jumps in protection levels correspond to changes in the number of satellites, and are also seen in the VDOP and HDOP plots shown previously in Figure 26.
Figure 29: Vertical protection level performance for modified LAAS and DFD1 (modified CNMP) architectures, entire King Air flight test
Figure 30: Horizontal protection level performance for modified LAAS and DFD1 (modified CNMP) architectures, entire King Air flight test

The results for position error and protection levels are summarized in Table 10 below. As mentioned, the largest DFD1 protection level approximations are used.

Table 10: Summary of results for entire flight

<table>
<thead>
<tr>
<th>Measure</th>
<th>Modified LAAS Architecture</th>
<th>DFD1 Architecture</th>
<th>Error Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vertical accuracy (95%), m</td>
<td>0.80</td>
<td>0.32</td>
<td>60.4%</td>
</tr>
<tr>
<td>Horizontal accuracy (95%), m</td>
<td>0.41</td>
<td>0.33</td>
<td>19.4%</td>
</tr>
<tr>
<td>Vertical protection level, m</td>
<td>5.90</td>
<td>2.60</td>
<td>59.6%</td>
</tr>
<tr>
<td>Horizontal protection level, m</td>
<td>3.24</td>
<td>1.38</td>
<td>57.4%</td>
</tr>
</tbody>
</table>
A Closer Look at Composite Protection Levels and Observed Error

In the DFD1 case, the mean value of the exact VPL is 2.598 m, and that of the VPL approximation is 2.599 m. These are almost identical to each other because the noise variance of the vertical error is much smaller than the total conspiring vertical bias, and therefore only one side of the univariate distribution extends outside the protection level in any calculable way. Thus the exact method, whose primary contribution is to reduce the PL by iteratively expanding the integration limits symmetrically outward from the origin instead of from the bias value, does not find any significant probability in the second tail of the distribution (the reader is referred to Figure 5). Accordingly, the improvement seen above in the VPL is almost entirely due to the reduction in noise achieved through the CNMP algorithm rather than the use of exact composite statistical methods. In the horizontal case, the spread between the exact HPL (1.11 m) and its approximations (1.31 m for #1 and 1.38 m for #2) is greater. This is because the horizontal approximation methods differ in how they calculate the bias, not the noise.

The reasonableness of conventionally calculated PL values may be subjected to a quick check by comparing the PL to the appropriate multiple of the standard deviation of the error bounded by the PL. In the LAAS results shown above, the vertical error is 0.80 m (95%), which corresponds approximately to the two-sigma level. The mean VPL for the six-sigma case is calculated by LAAS algorithms—assuming no bias and using a priori noise models—as 5.80 m. This is well above the corresponding quick check value, which is computed as $3 \times 0.80 = 2.40$ m. If anything, the quick check might indicate that VPL is too large and thus overly conservative, but is surely not too small.
In the DFD1 case, however, this simple check is inappropriate because it overlooks the relative magnitudes of the bias and the noise. The CNMP algorithm eventually reduces noise sigmas typically to 1 to 3 cm, while the range domain bias bound model includes minimum antenna group delay values of 0.10 m at the reference station and 0.20 m in the air, in addition to other bias components. Thus, in forming a composite PL using the CNMP algorithm, the bias typically dominates the noise.

For error conditions of this type, the DFD1 position domain error space can be visualized as a small noise ellipsoid at the end of a relatively long bias vector that extends well away from the origin. Thus, a composite PL may be quite close to a measured error value and still be valid. For example, a six-sigma composite VPL of 31 cm may be formed from a vertical position domain upper bias bound of +25 cm and a six-sigma noise range of ±6 cm. An observed two-sigma error of +27 cm would appear uncomfortably close to the edge of the protected VPL range of ±31 cm, but would not necessarily violate the VPL’s statistical integrity. However, it might indicate the need to perform a more comprehensive error analysis that incorporates confidence intervals, particularly if the size of the data set is relatively small compared to the probability level used to create the PL. The errors in this analysis are nowhere near as close to their PLs as in this fictitious example, and further analysis is left to future research.

Results for Six Approaches

The improvement in position error performance during the six 150-s approaches is also dramatic, as shown in Figure 31 and Table 11 for vertical error, and Figure 32 and Table 12 for horizontal error. The DFD1 architecture exhibits smoother behavior and
lower error than the LAAS baseline architecture according to every statistical measure. Most 95% error values are a little smaller than the values of $|\mu| + 2\sigma$, perhaps indicating that smoothing distorts otherwise normally distributed error. The slight upward trends in the DFD1 vertical error may be due to ionospheric and tropospheric decorrelation that decreased as the aircraft approached the runway threshold. The 150-s approach segments used in statistical analysis varied in length between 8 and 9.5 km, depending on aircraft speed. Only the final 8 km segments are shown below.

Figure 31: Vertical NSE performance for the LAAS architecture (top) and the DFD1 architecture (bottom), six 150-s approaches
Table 11: Vertical NSE statistics for six approaches (all values in m)

<table>
<thead>
<tr>
<th>Arch.</th>
<th>Statistic</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAAS</td>
<td>Mean</td>
<td>0.0167</td>
<td>-0.0567</td>
<td>0.7446</td>
<td>0.2807</td>
<td>0.3856</td>
<td>0.0977</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>0.2106</td>
<td>0.3188</td>
<td>1.0653</td>
<td>0.3637</td>
<td>0.4489</td>
<td>0.2663</td>
</tr>
<tr>
<td></td>
<td>Std Dev</td>
<td>0.0971</td>
<td>0.1635</td>
<td>0.1537</td>
<td>0.0453</td>
<td>0.0316</td>
<td>0.1249</td>
</tr>
<tr>
<td></td>
<td>Rms</td>
<td>0.0982</td>
<td>0.1725</td>
<td>0.7602</td>
<td>0.2843</td>
<td>0.3869</td>
<td>0.1582</td>
</tr>
<tr>
<td></td>
<td>95%</td>
<td>0.1945</td>
<td>0.3022</td>
<td>1.0405</td>
<td>0.3432</td>
<td>0.4355</td>
<td>0.2563</td>
</tr>
<tr>
<td></td>
<td>$</td>
<td>\mu</td>
<td>+ 2\sigma$</td>
<td>0.2109</td>
<td>0.3837</td>
<td>1.0519</td>
<td>0.3714</td>
</tr>
<tr>
<td></td>
<td>DFD1</td>
<td>Mean</td>
<td>-0.1126</td>
<td>-0.2722</td>
<td>0.1956</td>
<td>0.2911</td>
<td>0.1401</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>0.1750</td>
<td>0.3253</td>
<td>0.2894</td>
<td>0.3474</td>
<td>0.1807</td>
<td>0.0947</td>
</tr>
<tr>
<td></td>
<td>Std Dev</td>
<td>0.0239</td>
<td>0.0284</td>
<td>0.0418</td>
<td>0.0416</td>
<td>0.0189</td>
<td>0.0193</td>
</tr>
<tr>
<td></td>
<td>Rms</td>
<td>0.1151</td>
<td>0.2737</td>
<td>0.2000</td>
<td>0.2941</td>
<td>0.1414</td>
<td>0.0481</td>
</tr>
<tr>
<td></td>
<td>95%</td>
<td>0.1637</td>
<td>0.3181</td>
<td>0.2704</td>
<td>0.3377</td>
<td>0.1699</td>
<td>0.0767</td>
</tr>
<tr>
<td></td>
<td>$</td>
<td>\mu</td>
<td>+ 2\sigma$</td>
<td>0.1604</td>
<td>0.3290</td>
<td>0.2792</td>
<td>0.3743</td>
</tr>
</tbody>
</table>

Figure 32: Horizontal NSE performance for the LAAS architecture (top) and the DFD1 architecture (bottom), six 150-s approaches
In keeping with the use of horizontal instead of lateral protection levels in this analysis, the horizontal errors are computed as the length of the hypotenuse between the east and north error vectors at each GPS epoch. Consequently, they are magnitudes and are all greater than zero. The plots in Figure 32 show only half the y-axis range as the vertical plots in Figure 31. The lateral protection levels are always less than or equal to these horizontal values, since they project horizontal error onto the cross-track direction of a runway that may not be aligned with the horizontal bias vector or the major axis of the error ellipse.

Standard deviations are formed from unbiased variance by dividing the sum of squares (minus the mean) by \( n - 1 \) before taking the square root. Rms values are formed by dividing the sum of squares simply by \( n \) as in the conventional rms definition. In each case, there are \( n = 150 \) GPS epochs in each approach.

<table>
<thead>
<tr>
<th>Arch.</th>
<th>Statistic</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAAS</td>
<td>Mean</td>
<td>0.1180</td>
<td>0.1454</td>
<td>0.2977</td>
<td>0.1439</td>
<td>0.2055</td>
<td>0.1921</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>0.1948</td>
<td>0.1819</td>
<td>0.4488</td>
<td>0.1755</td>
<td>0.2436</td>
<td>0.2843</td>
</tr>
<tr>
<td></td>
<td>Std Dev</td>
<td>0.0503</td>
<td>0.0242</td>
<td>0.0474</td>
<td>0.0196</td>
<td>0.0279</td>
<td>0.0691</td>
</tr>
<tr>
<td></td>
<td>Rms</td>
<td>0.1282</td>
<td>0.1474</td>
<td>0.3014</td>
<td>0.1452</td>
<td>0.2074</td>
<td>0.2041</td>
</tr>
<tr>
<td></td>
<td>95%</td>
<td>0.1828</td>
<td>0.1739</td>
<td>0.3993</td>
<td>0.1722</td>
<td>0.2388</td>
<td>0.2788</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>\bar{u}</td>
<td>+ 2\sigma)</td>
<td>0.2186</td>
<td>0.1938</td>
<td>0.3925</td>
<td>0.1831</td>
</tr>
<tr>
<td>DFD1</td>
<td>Mean</td>
<td>0.1066</td>
<td>0.0329</td>
<td>0.1256</td>
<td>0.0932</td>
<td>0.1585</td>
<td>0.0656</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>0.1254</td>
<td>0.0499</td>
<td>0.1664</td>
<td>0.1249</td>
<td>0.1855</td>
<td>0.0768</td>
</tr>
<tr>
<td></td>
<td>Std Dev</td>
<td>0.0059</td>
<td>0.0077</td>
<td>0.0181</td>
<td>0.0115</td>
<td>0.0144</td>
<td>0.0041</td>
</tr>
<tr>
<td></td>
<td>Rms</td>
<td>0.1068</td>
<td>0.0338</td>
<td>0.1269</td>
<td>0.0939</td>
<td>0.1591</td>
<td>0.0658</td>
</tr>
<tr>
<td></td>
<td>95%</td>
<td>0.1183</td>
<td>0.0442</td>
<td>0.1544</td>
<td>0.1136</td>
<td>0.1821</td>
<td>0.0727</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>\bar{u}</td>
<td>+ 2\sigma)</td>
<td>0.1185</td>
<td>0.0483</td>
<td>0.1619</td>
<td>0.1162</td>
</tr>
</tbody>
</table>
Summary statistics for vertical and horizontal errors for both architectures, and the percent change between the architectures, are provided in Table 13. Here the overall combined mean of the approaches have been obtained simply by averaging the means of all the approaches. This is not meant to “average out” errors across approaches, but rather to indicate any bias that may persist across all the approaches. Six approaches are not sufficient for a statistically sound inference on this hypothesis, but they do serve as an initial indication of the validity of the DFD1 processing architecture. The combined maximum value is the maximum of all GPS epochs over all approaches. Assuming for the moment a common variance between all six approaches, the combined standard deviation is derived as the square root of the pooled estimator for this common variance, as in [100], dividing by \( m - 1 \) with \( m = 6 \) approaches. The combined 95% and \(|\hat{\mu}| + 2\sigma\) statistics are formed in the same way. The combined rms statistic is formed by applying the conventional rms formula to the rms values for each approach and dividing by \( m \).

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Statistic</th>
<th>LAAS</th>
<th>DFD1</th>
<th>Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Vertical</strong></td>
<td>Mean</td>
<td>0.24</td>
<td>0.03</td>
<td>87 %</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>1.07</td>
<td>0.35</td>
<td>67 %</td>
</tr>
<tr>
<td></td>
<td>Std Dev</td>
<td>0.13</td>
<td>0.03</td>
<td>73 %</td>
</tr>
<tr>
<td></td>
<td>Rms</td>
<td>0.38</td>
<td>0.20</td>
<td>48 %</td>
</tr>
<tr>
<td></td>
<td>95%</td>
<td>0.56</td>
<td>0.26</td>
<td>61 %</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>\hat{\mu}</td>
<td>+ 2\sigma)</td>
<td>0.59</td>
</tr>
<tr>
<td><strong>Horizontal</strong></td>
<td>Mean</td>
<td>0.18</td>
<td>0.10</td>
<td>47 %</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>0.45</td>
<td>0.19</td>
<td>59 %</td>
</tr>
<tr>
<td></td>
<td>Std Dev</td>
<td>0.13</td>
<td>0.03</td>
<td>73 %</td>
</tr>
<tr>
<td></td>
<td>Rms</td>
<td>0.20</td>
<td>0.11</td>
<td>47 %</td>
</tr>
<tr>
<td></td>
<td>95%</td>
<td>0.28</td>
<td>0.14</td>
<td>55 %</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>\hat{\mu}</td>
<td>+ 2\sigma)</td>
<td>0.30</td>
</tr>
</tbody>
</table>
RAIM Exclusions and Antenna Performance

RAIM exclusion of a single satellite occurred five different times during airborne position domain smoothing. Satellites 13 and 26 were each excluded twice, and satellite 6 once. In each case, the satellite was within 3 degrees of the 5-degree mask angle. The carrier phase propagation residuals that exceeded the 0.2 m threshold for these satellites were 0.49, 1.53, 1.48, 0.41, and 0.27 meters. The RAIM feature is essentially intended to be a safety mechanism of last resort and exceedingly rarely, if ever, used. Accordingly, checking receiver flags more thoroughly, tightening satellite pre-screening criteria and tuning the CNMP algorithm are all appropriate possibilities for future work in this area.

All of the performance criteria documented using the flight test data described here are well within GSL-D limits. Furthermore, the performance improvements listed above are for one dual-frequency GPS survey antenna that is not specialized for LAAS use. With proper siting of multiple antennas, and with reductions of code noise and multipath delivered by the CNMP algorithm, such an antenna type may achieve performance on par with the Integrated Multipath Limiting Antenna (IMLA) described in [114]. In antenna selection, however, other considerations besides multipath must also be taken into account. For example, antenna gain can be a key requirement to provide sufficient signal strength during ionospheric disturbances.
CHAPTER 9: SUMMARY AND CONCLUSIONS

Summary

A statistical characterization of composite horizontal protection levels VPL<sub>c</sub> and HPL<sub>c</sub> under the H0 hypothesis is provided, consistent with three key assumptions:

1. A certain amount of range domain bias is present in user measurements in normal GPS operation, without necessarily rejecting the H0 hypothesis.

2. Range domain error for the \( i \)-th satellite is modeled as an uncorrelated Gaussian random variable with non-zero mean \( \mu_i \) and variance \( \sigma_i^2 \), in a succession of probability distributions that are piecewise continuous. The bias represented by the mean is constant within a subdomain, but may vary between subdomains. This is also true of the variance. The bias is estimated by \( \hat{B}_i(t) \) and is bounded by steadily decreasing parameter \( B_{\text{max},i} \). Bias bounds, but not necessarily bias values, always conspire between satellites to produce the worst-case position domain value for protection level computation.

3. The variance-covariance matrix for the GPS position domain errors is derived from the range domain characterization, and the position domain error is modeled as multivariate normal with non-zero mean.

Based on these assumptions, composite protection levels VPL<sub>c</sub> and HPL<sub>c</sub> are implemented as univariate normal distributions with non-zero means. A simple method is presented by which exact values—that is, values accurate to a user-defined error tolerance and consistent with statistical assumptions—of VPL<sub>c</sub> and HPL<sub>c</sub> are obtained,
and by which computationally efficient approximations may be evaluated. A statistical quadratic form under the multivariate normal distribution is then used to derive a new class of protection levels based on the probability enclosed within a radius defined in two or more dimensions. A central chi-square representation of this quadratic form is also presented, and is incorporated into a six-step computational procedure for two-dimensional composite radial protection level $RPL_c$. This procedure is extended to the three-dimensional composite $SPL_c$ and $EPL_c$, and can be used for the fourth dimension of time as well.

Computationally efficient approximations for $VPL_c$, $HPL_c$ and $RPL_c$ are obtained. Results from a Monte-Carlo simulation for a single GPS measurement epoch are presented to illustrate these protection levels and their approximations. It is shown that existing methods for biased measurements are approximations of the exact methods shown here. In the case of $VPL_c$ and $HPL_c$, these approximations are shown to be overbounds of the exact values.

The improved performance offered by the DFD1 architecture is demonstrated against a LAAS baseline architecture using recorded flight test data from Ohio University’s King Air aircraft and reference station. Both architectures use only one reference antenna-receiver pair. The data set is 1 hour and 46 minutes in length, and includes six 150-s ILS aircraft approaches.

Distinctive elements of the DFD1 architecture implementation include the following:
• Use of the H0 hypothesis for integrity analysis, backed up by a mid-value select concept using three antenna-receiver pairs

• Employment of a single reference receiver antenna not specialized for use in LAAS

• Extension of the dual-frequency CNMP algorithm to estimate bias bounds based on maximum and minimum values of CMC observables in real time, instead of preset values

• Use of a low-pass filter with the CMC observables to further reduce noise and multipath prior to finding the maximum and minimum values

• Use of the CNMP algorithm in both reference station and airborne system

• Implementation of a floor bias of 0.10 m in the reference station and 0.20 m in the airborne system to account for antenna group delay errors

• Definition of new sigma curves separate from multipath effects and reduced by extended averaging

• Application of carrier phase position domain smoothing in the aircraft using bias and noise parameters to be transmitted from the reference station

• Weighting of the least-squares position solution and protection level computation by the inverse of the maximum bias bound

• Calculation of composite vertical and horizontal protection levels using a slightly more generic formulation than that specified for the LAAS approach service
Conclusions

The CNMP and CPDS algorithms in the DFD1 architecture significantly enhance accuracy and protection levels for differential GPS aircraft precision approach and landing operations. The exchange of code noise and multipath is seen to be a profitable one, with the final bound on introduced pseudorange bias much smaller than the original errors.

Results over the entire data set show an improvement in vertical accuracy from 0.80 m to 0.32 m (95%), and in horizontal accuracy from 0.41 m to 0.33 m (95%), reductions by 60% and 19%, respectively. During the six 150-s approaches, the mean vertical error is reduced from 0.24 m to 0.03 m, the maximum absolute vertical error from 1.07 m to 0.35 m, the standard deviation from 0.13 m to 0.03 m, the rms error from 0.38 m to 0.20 m, and the 95% error from 0.56 m to 0.28 m. Similarly, the mean horizontal error magnitude is reduced from 0.18 m to 0.10 m, the maximum horizontal error magnitude from 0.45 m to 0.19 m, the standard deviation from 0.13 m to 0.03 m, the rms error from 0.20 m to 0.11 m, and the 95% error from 0.28 m to 0.14 m.

The $|\mu| + 2\sigma$ NSE performance of 0.28 m vertical and 0.14 m horizontal shows that the DFD1 architecture attains the same levels of accuracy over the six approaches as any of the differential kinematic carrier phase architectures described in chapter 2 and listed in Table 2, at least for short differential baseline distances. It also achieves the same level of performance as the relative kinematic carrier phase architecture tested in aircraft carrier landings in [79], albeit in a more benign reference station multipath environment. If placed in an even lower multipath air-to-air relative environment, it may
possibly approach the levels of accuracy of the air-to-air refueling configurations of [51, 87-90].

The results of this analysis lend support to the possible use of dual-frequency receivers and CNMP algorithms at the reference station and the airborne system, non-specialized antennas at the reference station, CPDS algorithms in the airborne system, a composite approach to protection levels, and weighting by the inverse of the maximum bias bound. The explicit treatment of both noise and conspiring bias is seen to yield significant conceptual and practical benefit for aircraft precision approach and landing and similar applications.

**Resolution of Research Questions**

Based on the conclusions described above, the three research questions are answered as follows:

1. Protection levels *can* be computed exactly—that is, to an arbitrary specified error tolerance—for a set of non-zero mean, multivariate normal random variables that represent GPS position domain error. Approximations available for these protection levels have largely known error characteristics and may also be used.

2. A modified differential GPS processing architecture *does* significantly reduce position error and protection levels in flight tests compared to a conventional LAAS architecture in similar conditions. All measures of error computed for the precision approaches are reduced by at least 50%.

3. A modified differential GPS processing architecture that does not employ carrier phase integer ambiguity resolution *does* attain the same level of accuracy
performance as kinematic carrier phase processing differential and relative architectures that use dual-frequency carrier phase measurements and full ambiguity resolution in a surface-to-air role, but do not use inertial data.

The DFD1 processing architecture performs well in all three comparisons offered—to existing protection levels; to a conventional LAAS architecture in similar conditions; and to dual-frequency, real-time kinematic carrier phase systems—while using C/A code phase and dual-frequency carrier phase processing that does not require integer ambiguity resolution techniques. Thus, the DFD1 architecture constitutes a significant step forward for high-accuracy, high-integrity differential and relative GPS applications.
CHAPTER 10: FUTURE WORK

A great deal of fruitful work can still be accomplished in this research area. The following tasks follow naturally from the preceding results and conclusions:

1. Further characterize probability distribution(s) of observed multipath signatures, in order to reduce their error contributions.

2. Examine additional techniques to remove residual error due to ionospheric spatial decorrelation over long separation distances.

3. Estimate the worldwide availability of the DFD1 processing architecture under various conditions.

4. Improve the RAIM algorithm so that satellites are excluded for a specified interval and then readmitted for consideration.

5. Characterize the bounding performance of RPLc.

6. Investigate the possibility of additional performance gains from weighting by $1 / (\text{bias bound})^2$ instead of $1 / (\text{bias bound})$.

7. Check receiver flags more thoroughly, tighten satellite screening criteria and tune the CNMP algorithm, including optimizing the CNMP observation window size.

8. Implement and test a reference station with a mid-value select algorithm.

9. Examine the tradeoff between integrity gains from a receiver mid-value select architecture versus reduction in noise for a receiver averaging architecture, particularly focusing on measurement correlation between receivers.

10. Examine the statistical dependence between biases at the reference station and those in the airborne system. If dependence is found, calculate the probability that
these will simultaneously occur along with the worst-case noise values at both locations

11. Perform an integrity analysis of the DFD1 processing architecture, including measurement consistency monitoring at the reference station and cycle slip and loss of lock detection for all GPS receivers.

12. Test a DFD1-like architecture modified for air-to-air relative applications.

13. Examine composite PLs versus observed errors in more detail, deriving a confidence interval for the observed errors and comparing it to the PL.
REFERENCES


39. Jiyun Lee, Sam Pullen, Young Shin Park, Per Enge and Mats Brenner, “Position-Domain Geometry Screening to Maximize LAAS Availability in the Presence of


Proceedings of the IEEE Position Location and Navigation Symposium (PLANS)

82. Shuwu Wu et al., “Geometry Extra-Redundant Almost Fixed Solutions: A High
Integrity Approach for Carrier Phase Ambiguity Resolution for High Accuracy
Relative Navigation,” Proceedings of the IEEE/ION Position Location and
568-582.

83. Paul Sousa and Lee Wellons, “Results on Shipboard Relative GPS (SRGPS)
Testing for the Pegasus X-47A Unmanned Combat Air Vehicle (UCAV), Institute
of Navigation: Proceedings of the ION Annual Meeting/CIGTF 22nd Guidance

84. Steven Langel, Samer Khanafseh, Fang-Cheng Chan and Boris Pervan, “Cycle
Ambiguity Reacquisition in UAV Applications Using a Novel GPS/INS
Integration Algorithm,” Institute of Navigation: Proceedings of ION ITM-2009,
Anaheim, California, January 26-28, 2009.

85. Tim Katanik et al., “Interoperability Between Civil LAAS and Military JPALS
Precision Approach and Landing Systems,” Institute of Navigation: Proceedings

86. Bruce Peterson et al., “Investigation of Common Architectures for Land- and Sea-
Based JPALS,” Institute of Navigation: Proceedings of ION GNSS-2005, Long
Beach, California, September 13-26, 2005, pp. 26-37.


101. Alan Genz, “MVNLPS: Multivariate Normal Distribution Value for an
Ellipsoid,” Matlab implementation of Sheil and O'Muircheartaigh Algorithm AS
106, Washington State University, undated, available online at
http://www.math.wsu.edu/faculty/genz/homepage.

102. David Hsu, *Spatial Error Analysis: A Unified Application-Oriented Treatment*,
Wiley-IEEE Press, 2006, with companion Matlab software available for


pdf.

105. Sisir Vuyyuru, “Data Collection Network and Data Analysis for the Prototype
Local Area Augmentation System Ground Facility,” M.S.E.E. Thesis, School of
Electrical Engineering and Computer Science, Ohio University, November 2007.

106. Curtis Shively and Christopher Varner, “Impact of Carrier Smoothing on Event
Probabilities in LAAS,” Institute of Navigation: Proceedings of the ION Annual
Meeting, Albuquerque, New Mexico, June 24-26, 2002, pp. 195-205.

107. H. Euler and C. Goad, “On Optimal Filtering of GPS Dual Frequency
Observations Without Using Orbit Information,” Bulletin Geodesique, Vol. 65,


APPENDIX A: SOFTWARE FOR PROTECTION LEVEL SIMULATION

This appendix contains listings for two Matlab software files. The first is the function `calculate_protection_levels.m`, which was used to compute all protection levels in this dissertation. The second is the function `compositePLdemo.m`, which serves as a demonstration and test script that calls the first file. This demo file was used for all the simulation data and most of the plots presented in chapter 4.
calculate_protection_levels.m

function [VPL_H0,HPL_H0,RPL_H0,C,S,D,bias_signs,H_proj_vector,...
    biases_east,biases_north] = ...
    calculate_protection_levels (Hmat,svids,numvis,mu_ref,sig_ref,...
    mu_air,sig_air,theta_deg,K_ffmd,P_hmi,PL_method,LS_weighting_type)

% calculate_protection_levels: Determines VPLc, HPLc and RPLc
% function [VPL_H0,HPL_H0,RPL_H0,C,S,D,bias_signs,H_proj_vector,...
%     biases_east,biases_north] = ...
%     calculate_protection_levels (Hmat,svids,numvis,mu_ref,sig_ref,...
%     mu_air,sig_air,theta_deg,K_ffmd,P_hmi,PL_method,LS_weighting_type)
%
% This function calculates error protection levels for aircraft precision
% approach and landing and similar operations. Methods 4 and 6 are exact
% to the specified tolerance and in accordance with assumptions made in the
% references listed. Methods 2, 3 and 5 are computationally more efficient
% approximations to these methods. Code is reproduced in various methods
% to make for easier reading and comparison; a more modular approach would
% be used for implementation.
%
% Input parameters:
%   Hmat                Nx1     Matrix of unit vectors user-to-satellite
%   svids               Nx1     PRN code numbers identifying each GPS SV
%   numvis              1x1     Number of GPS satellites in the solution
%   mu_ref              Nx1     Biases at reference station, m
%   sig_ref             Nx1     Noise sigmas at reference station, m
%   mu_air              Nx1     Biases at aircraft, m
%   sig_air             Nx1     Noise sigmas at aircraft, m
%   theta_deg           Nx1     Satellite elevation angles, degrees
%   K_ffmd              1x1     Sigma multiplier: fault-free missed detections
%   P_hmi               1x1     Probability that pilot will see hazardously
%                               misleading information
%   PL_method           1x1     Type of protection level; see below
%   LS_weighting_type   1x1     0=unweighted least squares, 1 = weighted by
%                               1/bias, 2 = weighted by 1/variance
%
% Output parameters
%   VPL_H0              1x1     vertical +/- dist enclosing prob = 1 - Phmi, m
%   HPL_H0              1x1     horiz +/- dist enclosing prob = 1 - Phmi, m
%   RPL_H0              1x1     radial distance enclosing prob = 1 - Phmi, m
%   C                   4x4     Position domain correlation matrix (x,y,z,t)
%   S                   4xN     Projection matrix, range to pos'n domains
%   D                   4x4     DOP matrix
%   bias_signs          1xN     Signs of SV biases yielding worst case error
%   H_proj_vector       2x1     Unit vector defining noise ellipse semi-major
%                       axis
%   biases_east         1x2^numvis  Position domain east biases available
%                       (Used only for simulation plots)
%   biases_north        1x2^numvis  Position domain north biases available
%                       (Used only for simulation plots)
%
% Methods: (COMP means "Composite PL"):
% 1 - LAAS (no explicit bias term); LS weighted by 1/variance
% 2 - COMP  VPLc approximation 1, HPLc approximation 1:
Takes absolute value of S matrix elements and biases, then combines with noise. Lower excess overbound than HPL approx 2. Similar to HPL_H1 method in LAAS.

3 - COMP VPLc approximation 1, HPLc approximation 2: For horiz case, stretches out biases to be collinear, then adds collinearly with semi-major noise axis. From FvG/TAS; good for single bias value.

4 - COMP Exact VPLc and RPLc (i.e., exact to specified prob error tolerance and with assumptions of non-zero mean multivariate normal for position domain errors): Integrates univariate inverse normal distributions for vertical and horizontal to find tightest PLs in presence of bias. Yields same results as 2 and 3 when no bias exists, but provides the smallest overbound when bias does exist.

5 - COMP RPLc Approximation 1: uses same short bias search as in 4, then applies integration of bivariate normal to find approximate radial protection level RPLc. Not guaranteed to be an overbound (without further research). Does not calculate VPLc or HPLc.

6 - COMP Exact RPLc (exact in same sense as # 4 above). Performs full search for bias combination that, together with horiz noise error ellipse, produced smallest RPLc that contains desired percent of errors. Prohibitively slow for very small P_hmi values and number of SVs more than 6; intended as a baseline to evaluate other methods. May easily be extended to higher dimension(s).

Note also: When biases are significant compared to noise sigma, unweighted LS or weighting by 1/bias should be used, as the weighting by 1/variance will make things worse as bias gets larger relative to noise sigma. Eventually the PL will not enclose the required probability without it.

References:
A. Genz, algorithm MVNLPS (see this file, at bottom). This and other free Matlab statistical software is available for download from http://www.math.wsu.edu/faculty/genz/software/software.html
This author's works are primary references for the Matlab Statistics toolbox. Algorithm MVNLPS reproduced here by permission of the author.
if nargin < 12
    LS_weighting_type = 0; % 0 = no (default)
end

if numvis < 4
    PL_method = -1; % Shouldn't do PL if number of SVs < 4
end

% Initialize parameters
VPL_H0 = 99.9;
HPL_H0 = 99.9;
RPL_H0 = 99.9;
H_proj_vector = []; % Calculated only in methods 2-4
bias_signs = ones(1,numvis); % Calculated only in methods 4-6
biases_east = []; % Calculated only in methods 4-6
biases_north = [];
sigma_total = sqrt ( sig_ref.^2 + sig_air.^2 ); % Differential system noise
mu_total = sqrt ( mu_ref.^2 + mu_air.^2 ); % Differential system biases

if PL_method == 1 % LAAS, no explicit bias term, use weighted LS always
    % This method gives correct answers only for zero bias. If bias
    % is added, this method will fail. However, the H0 hypothesis would be
    % rejected and the H1 hypothesis would take over.
    Wqr = diag(1 ./ sigma_total );
    % modified LS weighting matrix, units of 1/m (not 1/m^2). This matrix
    % is used for QR factorization, and is equivalent to the WLS used
    % in the LAAS MASPS/WAAS MOPS.

    % Use QR factorization for stable calculation of S matrix
    % Note that the weight matrix Wqr has 1/sigma on the diagonals
    % (not 1/sigma.^2 as does the W matrix in the WAAS MOPS)
    [Q, R] = qr(Wqr*Hmat); Qt = Q';
    Qx = Qt(1:4,1:numvis);
    U = R(1:4,1:4);
    S = inv(U)*Qx*Wqr;
    % S is matrix of partial derivatives that translate range domain
    % vectors to position domain vectors for each SV
    % Alternate WLS method from sec 2.3.9 of WAAS MOPS; not as stable
    % but shown to be equivalent to QR method on 6/4/2008; DCB
    W = inv(diag(sigma_total.^2));
    S = inv(Hmat'*W*Hmat) * Hmat' * W;

    T = S * diag(sigma_total); % Weighted DOP matrix
    C = T * T'; % Covariance matrix not used in LAAS here but is
% calculated here to return to calling function
D = S * S';
% Could also call Calculate_cov_matrix.m

% Calculate VPL_H0 and HPL_H0 values. This is a vectorized version;
% This also may be done recursively for illustration purposes.

sigR = 1./diag(Wqr);% (numvis x 1);Wqr contains 1/sigma not 1/(sigma^2)

D = S * S';% Calculate VPL_H0 and HPL_H0 values. This is a vectorized version;
% This also may be done recursively for illustration purposes.

sigR_sq = sigR.^2; % (numvis x 1)

% Calculate VPL_H0 and HPL_H0 values. This is a vectorized version;
% This also may be done recursively for illustration purposes.

S1_sq = S(1,:).^2; % (1 x numvis)
S2_sq = S(2,:).^2; % (1 x numvis)
S3_sq = S(3,:).^2; % (1 x numvis)

sigma_sq_up = S3_sq * sigR_sq;

% Calculate VPL_H0 and HPL_H0 values. This is a vectorized version;
% This also may be done recursively for illustration purposes.
p1 = ( S1_sq + S2_sq ) * sigR_sq;
p2 = ( S1_sq - S2_sq ) * sigR_sq;
p3 = ( S(1,:) .* S(2,:) ) * sigR_sq; % d_EN in WAAS/d_xy in LAAS TAP,
% equivalent to cov(X,Y), the covariance of RV's X and Y

% Calculate VPL_H0 and HPL_H0 values. This is a vectorized version;
% This also may be done recursively for illustration purposes.
theta1 = 0.5*atan2(2*p3,p2);
% CCW positive angle in horizontal plane between east axis and
% projection of semi-major axis of 3-D noise ellipse onto horizontal
% plane, adapted from Torrieri using atan2 and larger function range,
% here -pi/2 < theta1 < pi/2 (the 0.5 multiple reduces that -pi to pi
% range of atan2 by half) Equivalent to the following:
% sigma_east  = sqrt(C(1,1)); sigma_north = sqrt(C(2,2));
% rho_xy = p3 / ( sigma_east * sigma_north ); % Corr coeff
% theta1 = 0.5 * atan2( 2 * rho_xy * sigma_east *
% sigma_north,...
% (sigma_east^2 - sigma_north^2) );

% Calculate VPL_H0 and HPL_H0 values. This is a vectorized version;
% This also may be done recursively for illustration purposes.

VPL_H0_sigma = sqrt(sigma_sq_up);

% Calculate VPL_H0 and HPL_H0 values. This is a vectorized version;
% This also may be done recursively for illustration purposes.

VPL_H0 = K_ffmd * VPL_H0_sigma; % Composite VPL

HPL_H0_sigma = sqrt( 0.5*p1 + sqrt( (0.5*p2)^2 + p3^2) ); % 1 sigma
% value along semimajor axis due to noise etc. but not bias

% Calculate VPL_H0 and HPL_H0 values. This is a vectorized version;
% This also may be done recursively for illustration purposes.
HPL_H0 = K_ffmd * HPL_H0_sigma; % Composite HPL

elseif PL_method == 2 % Composite PL approximation 1 ("standard" HPL overbound)
% Uses absolute value of S matrix elements and biases; similar to
% calculation of HPL_H1 in LAAS MASPS; Not physically realizable because
% changing the sign of an S element in the row 1, col i (x) and a not
% changing the sign of S(2,i) implies that the same satellite can be
% reciprocal bearings from the user at the same time, which is
% impossible in GPS
% Here propagation of sigma for the noise component is modeled.
% If noise variance differs from one satellite to another, I must model
% that in the covariance matrix for my PL calculation even if use
% unweighted LS. The simple fact that the noise is not uniformly the
% same allows it to propagate through the solution in a non-uniform
% way, and must be accounted for. Note: if all the weighting parameters

% Calculate VPL_H0 and HPL_H0 values. This is a vectorized version;
% This also may be done recursively for illustration purposes.

H_proj_vector = [cos(theta1) sin(theta1)]'; % projection of semi-major
% axis of 3-D noise ellipse onto horizontal plane
% are equal (1/bias or 1/variance), by default the method becomes
% unweighted LS regardless of the option selected.
[C,S,D] = Calculate_cov_matrix (Hmat, sigma_total, mu_total,...
  LS_weighting_type); % C,S weighted or unwt'd

VPL_H0_sigma = sqrt(S(3,:).^2 * sigma_total.^2);
% VPL_H0_sigma = sqrt(C(3,3)); % alternate form with same result

dx_sq = C(1,1);
dy_sq = C(2,2);
dxy = C(1,2);

p1 = ( dx_sq + dy_sq );
p2 = ( dx_sq - dy_sq );
p3 = ( dxy );

% 1 sigma value along semimajor axis due to noise etc. but not bias:
HPL_H0_sigma = sqrt( 0.5*p1 + sqrt( (0.5*p2)^2 + p3^2) );

theta1 = 0.5*atan2(2*p3,p2); % see note above
thetal1d = theta1 * 180/pi; % in degrees
H_projection_vector = [cos(theta1) sin(theta1) ]'; % projection of semi-major
% axis of 3-D noise ellipse onto horizontal plane

% Now model how the bias propagates thru to posn domain in posn
% solutions A worst-case combination of the x and y biases separately
% is assumed, and then they are combined. This is not physically
% realizable, as explained above. Other overbound methods exist.
VPL_H0_bias = abs(S(3,:,:)) * mu_total;
HPL_H0_bias = hypot(abs(S(1,:))*mu_total,abs(S(2,:))*mu_total);

% % For development: assume unweighted LS
% G = inv(Hmat' * Hmat) * Hmat';
% VPL_H0_bias = abs(G(3,:,:)) * mu_total;
% HPL_H0_bias = hypot(abs(G(1,:,:))*mu_total,abs(G(2,:,:))*mu_total);

VPL_H0 = K_ffmd * VPL_H0_sigma + VPL_H0_bias; % Composite VPL, overbound
HPL_H0 = K_ffmd * HPL_H0_sigma + HPL_H0_bias; % Composite HPL, overbound

elseif PL_method == 3 % Composite PL approximation 2 ("wide" HPL overbound)

% Stretches out all component bias vectors to be collinear; TAS/FvG method;
% Not physically realizable but convenient if all range domain bias bounds
% are the same, but bias term for each SV turns out to be bias * HDOP
% and is supported by some theoretical derivations by Lee, Loh & Fernow,
% van Graas and Skidmore.

% Model propagation of sigma for the noise component (see note above)
[C,S,D] = Calculate_cov_matrix (Hmat, sigma_total, mu_total,...
  LS_weighting_type); % C,S weighted or unwt'd

VPL_H0_sigma = sqrt(S(3,:).^2 * sigma_total.^2);
% VPL_H0_sigma = sqrt(C(3,3)); % alternate form with same result

dx_sq = C(1,1);
dy_sq = C(2,2);
dxy = C(1,2);

p1 = ( dx_sq + dy_sq );
\[ p2 = (dx^2 - dy^2); \]
\[ p3 = (dy); \]

% 1 sigma value along semimajor axis due to noise etc. but not bias:
HPL_H0_sigma = sqrt( 0.5*p1 + sqrt( (0.5*p2)^2 + p3^2) );

\[ \text{theta1} = 0.5*\text{atan2}(2*p3,p2); \] % see note above
\[ \text{theta1d} = \text{theta1} \times \frac{180}{\pi}; \] % in degrees
\[ H\_\text{proj}\_\text{vector} = \begin{bmatrix} \cos(\text{theta1}) & \sin(\text{theta1}) \end{bmatrix}; \] % projection of semi-major axis of 3-D noise ellipse onto horizontal plane

% Model how the bias propagates (see note above):
VPL_H0_bias = abs(S(3,:)) \times \mu\_total;
% The following line is different than solution above:
HPL_H0_bias = \sum \left( \sqrt{ (S(1,:)) \times \mu\_total'^2 + (S(2,:)) \times \mu\_total'^2} \right);)
% Equivalent to:
HPL_H0_bias = \sum(\text{hypot}( S(1,:), S(2,:), \mu\_total' ));

VPL_H0 = K\_ffmd \times VPL\_H0\_sigma + VPL\_H0\_bias; % Composite VPL, overbound
HPL_H0 = K\_ffmd \times HPL\_H0\_sigma + HPL\_H0\_bias; % Composite HPL, overbound

elseif PL\_method == 4 % "Exact" to prob error tol

% Radial method here not really workable for very small P\_hmi and especially bias > 10 * sigma.
prob\_level = 1 - P\_hmi;

% Use integration to find VPL (& for HPL, search for worst-case horiz bias)
% In functions below, if P\_hmi \leq N, then numterms > N and tol < \leq (N+1)
% must be observed.
sigma\_total = \sqrt{ sig\_ref'^2 + sig\_air'^2 } ; % Noise
mu\_total = \sqrt{ mu\_ref'^2 + mu\_air'^2 } ; %

% Model how the noise propagates:
[C,S,D] = Calculate\_cov\_matrix (Hmat, sigma\_total, mu\_total,...
% LS\_weighting\_type); % C,S weighted or unweighted
Cxy = C(1:2,1:2); % for RPL calculation
VPL\_H0\_sigma = sqrt(C(3,3));

\[ dx\_sq = C(1,1); \]
\[ dy\_sq = C(2,2); \]
\[ dxy = C(1,2); \]

\[ p1 = (dx\_sq + dy\_sq); \]
\[ p2 = (dx\_sq - dy\_sq); \]
\[ p3 = (dxy); \]

% 1 sigma value along semimajor axis due to noise etc. but not bias:
HPL\_H0\_sigma = sqrt( 0.5*p1 + sqrt( (0.5*p2)^2 + p3^2) );

\[ \text{theta1} = 0.5*\text{atan2}(2*p3,p2); \] % + is CCW from horiz x axis, from Torrieri
\[ \text{theta1d} = \text{theta1} \times \frac{180}{\pi}; \] % in degrees
\[ H\_\text{proj}\_\text{vector} = \begin{bmatrix} \cos(\text{theta1}) & \sin(\text{theta1}) \end{bmatrix}; \] % projection of semi-major axis of 3-D noise ellipse onto horizontal plane

% Model how bias propagates (see note above):
[C,S,D] = Calculate\_cov\_matrix (Hmat, sigma\_total, mu\_total,...
LS_weighting_type); % C,S weighted or unweighted

% Now model how the bias propagates thru posn solutions
%  G = inv(Hmat' * Hmat) * Hmat'; % for devel. assume unweighted method used

VPL_H0_bias = abs(S(3,:)) * mu_total; % vert offset of the ellipse
VPL_H0 = nf9(prob_level,VPL_H0_bias,VPL_H0_sigma);

% Create matrix of all possible sign combinations of uniform PR
% bias for a given number of SVs
ncom = 2^numvis; % number of signed bias combinations (+ or -)
bias_combns = ones(ncom,numvis);
for k1 = 1:numvis % number of columns
    ncy = 2^k1; % number of cycles of +/- signs in this column
    for k2 = 1:ncy
        bias_combns((k2-1)*ncom/ncy + 1 : k2*ncom/ncy , k1) = ones(ncom/ncy,1) * (-1)^(k2+1);
    end
    end

biases_east = bias_combns * (S(1,:)' .* mu_total);
biases_north = bias_combns * (S(2,:)' .* mu_total);
H_biases_projected = [biases_east biases_north] * H_proj_vector;
[maxhb,ii] = max(abs(H_biases_projected));
bias_signs = bias_combns(ii,:);
HPL_H0_bias = H_biases_projected(ii);

% Solve for HPL_H0 using integration for exact solution given the
% HPL_H0_bias method selected immediately above
HPL_H0 = nf9(prob_level,HPL_H0_bias,HPL_H0_sigma);

elseif PL_method == 5 % RPlc approximation 1 (no VPlc or HPlc computation)

prob_level = 1 - P_hmi;

% Use integration to find VPL (& for HPL, search for worst-case horiz bias)
% In functions below, if P_hmi = le-N, then numterms > N and tol < le-(N+1)
% must be observed.
sigma_total = sqrt ( sig_ref.^2 + sig_air.^2 ); % Noise
mu_total = sqrt ( mu_ref.^2 + mu_air.^2 );

% Model how the noise propagates:
[C,S,D] = Calculate_cov_matrix (Hmat, sigma_total, mu_total,...
    LS_weighting_type); % C,S weighted or unweighted
Cxy = C(1:2,1:2); % for RPl calculation
VPL_H0_sigma = sqrt(C(3,3));

dx_sq = C(1,1);
dy_sq = C(2,2);
dxy = C(1,2);

p1 = ( dx_sq + dy_sq );
p2 = ( dx_sq - dy_sq );
p3 = ( dxy );

% 1 sigma value along semimajor axis due to noise etc. but not bias:
HPL_H0_sigma = sqrt( 0.5*p1 + sqrt( (0.5*p2)^2 + p3^2) );

thetal = 0.5*atan2(2*p3,p2); % + is CCW from horiz x axis, from Torrieri
\theta_1d = \theta_1 \times 180/\pi; \% \text{in degrees}
H_{proj\_vector} = \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) \end{bmatrix}'; \% \text{projection of semi-major axis of 3-D noise ellipse onto horizontal plane}

% Create matrix of all possible sign combinations of uniform PR
% bias for a given number of SVs
ncom = 2^\text{numvis}; \% number of signed bias combinations (+ or -)
bias_combns = ones(ncom,\text{numvis});
for k1 = 1:\text{numvis} \% number of columns
    ncy = 2^\text{k1}; \% number of cycles of +/- signs in this column
    for k2 = 1:ncy
        bias_combns( ... 
            (k2-1)*ncom/ncy + 1 : k2*ncom/ncy , k1 ) = ... 
        ones( ncom/ncy , 1 ) * (-1)^{(k2+1)};
    end
end

biases_east = bias_combns * (S(1,:)' .* \mu_{total});
biases_north = bias_combns * (S(2,:)' .* \mu_{total});
H_{biases\_projected} = [biases_east biases_north] * H_{proj\_vector};
[maxhb,ii] = max(abs(H_{biases\_projected}));
bias_signs = bias_combns(ii,:);
RPL_{H0\_bias\_east} = biases_east (ii);
RPL_{H0\_bias\_north} = biases_north(ii);

% Find the RPL
RPL_{H0} = edcaci3(\text{prob\_level},[RPL_{H0\_bias\_east},RPL_{H0\_bias\_north}]',\text{Cxy});

elseif \text{PL\_method} == 6 \% Exact \text{RPLc} (no \text{VPLc} or \text{HPLc} computation)
    \% This method here not really workable for very small \text{P\_hmi},
    \% \text{numvis} > 8 and especially bias > 10 \times \sigma.
    \text{prob\_level} = 1 - \text{P\_hmi};

    \% Use integration to find \text{VPL} (& for \text{HPL}, search for worst-case horiz bias)
    \% In functions below, if \text{P\_hmi} = 1e-N, then \text{numterms} > N and \text{tol} < 1e-(N+1)
    \% must be observed.
    \sigma_{total} = \sqrt{ \text{sig\_ref}^2 + \text{sig\_air}^2 }; \% \text{Noise}
    \mu_{total} = \sqrt{ \text{mu\_ref}^2 + \text{mu\_air}^2 };

    \% Model how the noise propagates:
    [\text{C},\text{S},\text{D}] = \text{Calculate\_cov\_matrix} (\text{Hmat}, \sigma_{total}, \mu_{total}, ... 
        \text{LS\_weighting\_type}); \% \text{C},S weighted or unweighted
    \text{Cxy} = \text{C}(1:2,1:2); \% for \text{RPL} calculation

    \% Find \text{RPL} by brute force (search through all bias combinations)
    \text{h2} = \text{waitbar}(0,... 
        'Searching PR bias combinations for \text{RPL}. Please wait...');

    ncom = 2^\text{numvis}; \% number of signed bias combinations (+ or -)
    RPL_{H0} = 0;
    bias_combns = ones(ncom,\text{numvis});
    for k1 = 1:\text{numvis} \% number of columns
        ncy = 2^\text{k1}; \% number of cycles of +/- signs in this column
        for k2 = 1:ncy
            bias_combns( ... 
                (k2-1)*ncom/ncy + 1 : k2*ncom/ncy , k1 ) = ... 
            ones( ncom/ncy , 1 ) * (-1)^{(k2+1)};
        end
end
end

biases_east = bias_combns * (S(1,:)' .* mu_total);
biases_north = bias_combns * (S(2,:)' .* mu_total);

for k3 = 1:ncom

    % Find the RPL
    RPL_temp = edcaci3(prob_level,[biases_east(k3),biases_north(k3)]',Cxy);
    if abs(RPL_temp) > RPL_H0 % New max; save params
        RPL_H0 = RPL_temp;
        bias_signs = bias_combns(k3,:);
    end
    waitbar(k3/ncom,h2)
end

end % function Calculate_protection_levels

else

    error('PL_method must be integers 1 through 6')

end % if PL_method == -1

end % function Calculate_protection_levels

% ************************************************************************
% function [C,S,D] = Calculate_cov_matrix (Hmat, sigma_total, mu_total,...
% LS_weighting_type)
%
% Calculates position domain GPS error covariance matrix
%
% Inputs:
% Hmat              4xn     rows are [dE dN dU] unit vectors user to GPS
%                     SV, and 1 for time
% sigma_total       nx1     Total range domain noise, m
% mu_total          nx1     Total range domain bias, m
% LS_weighting_type  (0 = no wt; 1 = weight by 1/bias bound, 2 = wt by
%                     1/variance)
%
% Outputs:
% C                 4x4     Position domain covariance matrix
% S                 4xn     Projection matrix from range to posn domain
% T                 4x4     DOP matrix (C matrix without sigma.^2 values)
% BW                4x4     Bias weighting matrix, like covariance matrix
%
% Equivalent forms:
% % Covar matrix from unweighted S matrix has to add sigma in:
% cov_dPR = diag (sigma_total.^2); % Per V. Krishnan OU Thesis, App A
% S = inv(Hmat' * Hmat) * Hmat'; % Unweighted version, sometimes termed G
% C = ( S * cov_dPR * S' );
% C = zeros(2,2); % (simpler 2-D example)
% C(1,1) = S(1,:).^2 * sigma_total.^2;
% C(2,2) = S(2,:).^2 * sigma_total.^2;
% C(1,2) = ( S(1,:) .* S(2,:) ) * sigma_total.^2;
% C(2,1) = C(1,2);
%
% 2009-05-17 DCB Revised to use matrix T instead of cov_dPR
%
% First, account for how non-uniform noise (i.e., sigma varies from one
% satellite to another) propagates through the position solution.
% If all sigmas are equal, it becomes unweighted LS automatically,
% since all the weights are equal and cov_dPR will be an identity matrix
% times a constant.

\[
\text{cov}_{dPR} = \text{diag} \left( \sigma_{total}^2 \right) \; ; \; \% \text{ Per V. Krishnan OU Thesis 2004, App A}
\]

% Now account for any weighting that is performed in the position solution.
% Since bias propagates through the solution linearly instead of in a root-
% sum-square fashion, it does not need a correction factor such as \text{cov}\_dPR.

if \text{LS\_weighting\_type} == 0 \; \% \text{unweighted LS in pos}'n solution

\[
\text{numsv} = \text{length}(\sigma_{total}) \; ; \; \%
\text{Identity matrix -- no weighting}
\]

\[
\text{S} = \text{inv} \left( \text{Hmat}' \; \times \; \text{W} \; \times \; \text{Hmat} \right) \; \times \; \text{Hmat}' \; \times \; \text{W} \; ; \; \%
\text{Projection matrix}
\]

\[
\text{C} = \text{S} \; \times \; \text{cov}_{dPR} \; \times \; \text{S}' \; ; \; \%
\text{DOP matrix}
\]

end

elseif \text{LS\_weighting\_type} == 1 \; \% \text{Weighted by 1/bias bound}

if any(abs(\mu_{total}))<0.01 \; % \text{avoid matrix singularities}

\[
\text{mu}\_\text{total}(\text{find}(\text{abs}(\mu_{total})<0.01)) = 0.01;
\]

end

\[
\text{W} = \text{inv}(\text{diag}(\text{abs}(\mu_{total}))) \; ; \; \%
\text{Weighting matrix 1/bias}
\]

\[
\text{S} = \text{inv} \left( \text{Hmat}' \; \times \; \text{W} \; \times \; \text{Hmat} \right) \; \times \; \text{Hmat}' \; \times \; \text{W} \; ; \; \%
\text{Projection matrix}
\]

\[
\text{C} = \text{S} \; \times \; \text{cov}_{dPR} \; \times \; \text{S}' \; ; \; \%
\text{Covar matrix}
\]

\[
\text{D} = \text{S} \; \times \; \text{S}' \; ; \; \%
\text{DOP matrix}
\]

end

elseif \text{LS\_weighting\_type} == 2 \; \% \text{Weighted by 1/variance (LAAS)}

\[
\text{W} = \text{inv}(\text{diag}(\sigma_{total}^2)) \; ; \; \%
\text{Weighting matrix 1/variance}
\]

\[
\text{S} = \text{inv} \left( \text{Hmat}' \; \times \; \text{W} \; \times \; \text{Hmat} \right) \; \times \; \text{Hmat}' \; \times \; \text{W} \; ; \; \%
\text{Projection matrix}
\]

\[
\text{C} = \text{S} \; \times \; \text{cov}_{dPR} \; \times \; \text{S}' \; ; \; \%
\text{Covariance matrix}
\]

\[
\text{D} = \text{S} \; \times \; \text{S}' \; ; \; \%
\text{DOP matrix}
\]

else

\text{error} ('\text{LS\_weighting\_type must be 0, 1 or 2}')

end \%if \text{LS\_weighting\_type} == 0

end \% function

% ************************************************************************

function r = \text{n}\_9(p,mu,sig)
% \text{NF9}: Inverse prob. for Normal PDF \sim \text{N}(mu,sig)
%
function r = nf9(p,a,mu,sig)

% r = limit of 2-sided symmetrical interval [-r,r] that contains x with
% probability p
% p = probability that a point x chosen from n normally distributed
% random vector X ~ N(mu,sig) is within the interval [-r,r].
% mu = mean
% sig = standard deviation, sigma

% See also NF8, NF2, NF3

% This file by Dean C. Bruckner, August 28, 2008
% (c) 2008 by Dean C. Bruckner/Ohio University, all rights reserved
% 2008-08-28 DCB Initial version, adapted from EDCACI by D.Y. Hsu
% Addition to Spatial Error Analysis ToolBox Version 1.0, October 5, 1997
% (c) 1997-1998 by David Y. Hsu
if p<0 || p>1 || sig<0
    error('One or more input arguments out of range')
end

r=1; % Starting point
p0 = nf8(r,mu,sig);
numterms=9;% [assumes quadratic convergence; Hsu's orig value = 5]
for j=1:2:numterms
    while (p0 > p)
        r=r-(0.1)^j;
        p0=nf8(r,mu,sig);
    end
    while (p0 < p)
        r=r+(0.1)^j;
        p0=nf8(r,mu,sig);
    end
end

function p = nf8(a,mu,sig)
% NF8: Integrates univariate normal PDF ~N(mu,sig) over interval [-a,a]
% function p = nf8(a,mu,sig)
% a = input specifying limits [-a,a] of probability integral
% mu = mean of PDF
% sig = sigma of PDF
% p = probability that a point x chosen from a univariate normally
% distributed random vector X is within the interval [-a,a]
% See also NF2, NF6

% This file by Dean C. Bruckner, August 21, 2008
% (c) 2008 by Dean C. Bruckner/Ohio University, all rights reserved
% 2008-08-21 DCB Initial version
Addition to Spatial Error Analysis ToolBox Version 1.0, October 5, 1997
(c) 1997–1998 by David Y. Hsu

Ref: http://mathworld.wolfram.com/NormalDistributionFunction.html

\[ a_2 = \frac{a - \mu}{\sigma}; \quad \text{upper integration limit} \]
\[ a_1 = \frac{-a - \mu}{\sigma}; \quad \text{lower integration limit} \]

\[ p = 0.5 \left( \text{erf}(a_2/\sqrt{2}) - \text{erf}(a_1/\sqrt{2}) \right); \]

Equivalent to the following from the Matlab Statistics toolbox:
\[ \text{normcdf}(a_2, \mu, \sigma) - \text{normcdf}(a_1, \mu, \sigma) \]

*****************************************************************************

function R = edcaci3(p,mu,V)

EDCACI3: Inverse prob. for correlated elliptical PDF over offset circle

function R = edcaci3(p,sx,sy,h,k,V)

R = radius of offset circle with center at (0,0)
mu = [x,y]' column vector of coordinates of offset of ellipse
V = correlation matrix between x and y
p = probability that a point (x,y) chosen from correlated, bivariate
normally distributed random vectors \( X \sim N(\mu_x, sx) \), \( Y \sim N(\mu_y, sy) \)
with correlation coefficient rho is within the area of a circle with
center at (0,0).

See also EDCACD, EDCACI

This file by Dean C. Bruckner, March 28, 2008
(c) 2008 by Dean C. Bruckner/Ohio University, all rights reserved
2008-03-28 DCB Initial version, adapted from EDCACI by D.Y. Hsu

if p<0 || p>1
    error('Probability level p must be in range [0,1]')
end

R=1; % Starting point
p0 = edcacd3(R,mu,V);
numterms=9;% [assumes quadratic convergence; Hsu's orig value = 5]
for j=1:2:numterms
    while (p0 > p)
        R=R-(0.1)^j;
        p0=edcacd3(R,mu,V);
    end

    while (p0 < p)
        R=R+(0.1)^j;
        p0=edcacd3(R,mu,V);
    end
function p = edcacd3(R,mu,V)

% EDCACD3: Integrates correlated elliptical PDF over offset circular area
% function p = edcacd3(R,mu_x,mu_y,V)
% % R = radius of offset circle with center at (0,0)
% % mu = [x,y]' column vector of coordinates of offset of ellipse
% % V = correlation matrix between x and y
% % p = probability that a point (x,y) chosen from correlated, bivariate
% % normally distributed random vectors X ~ N(mu_x,sx), Y ~ N(mu_y,sy)
% % with correlation coefficient rho is within the area of a circle with
% % center at (0,0).
% % The covariance matrix for GPS observations is constructed as follows:
% % Construct the covariance matrix for East-North (see
% % derivation of SIGMA from matrix S, vector sigma_total and
% % Theorem 5-12 of ref 2 below
% V = zeros(2,2);
% V(1,1) = S(1,:).^2 * sigma_total.^2;
% V(2,2) = S(2,:).^2 * sigma_total.^2;
% V(1,2) = ( S(1,:) .* S(2,:) ) * sigma_total.^2;
% V(2,1) = V(1,2);
% where S is the 4 x N matrix of partial derivatives mapping the GPS range
% domain into the east/north/up/time domain, also called the position
% domain. We consider the range domain errors of the N satellites to be
% uncorrelated RVs U_i with distribution ~N(mu_r,sigma_r), where mu is the
% bias and sigma is the standard deviation of the noise. Matrix S
% deterministically maps mu and sigma from the range domain into position
% domain RVs X, Y, Z and T, which are also Normal distributions with mean
% and covariance as produced by Ref. 2 below. The position domain biases
% that result from this transformation (i.e., the expected value of X and
% of Y) are in vector mu. The covariance of X and Y is contained in V.
% See also EDCACI2, EDCAUD2, EDCAUD, EDCACI

% This file by Dean C. Bruckner, March 28, 2009
% (c) 2009 by Dean C. Bruckner/Ohio University, all rights reserved
% 2008-03-17 DCB Initial version. Adapted in part from EDCACD by D.Y. Hsu.
% Addition to Spatial Error Analysis ToolBox Version 1.0, October 5, 1997
% (c) 1997-1998 by David Y. Hsu

% Ref:
% 1. J. Sheil and I. O'Muircheartaigh, "Algorithm AS 106: The Distribu-
% tion of Non-Negative Quadratic Forms in Normal Variables", Applied
% the Royal Statistical Society, Stable URL:
% http://www.jstor.org/stable/2346884
if R<0
    error('R must be nonnegative.')
end

% Use Sheil and O'Muircheartaigh's Algorithm AS 106, per Genz
q = [0 0]'; % bias (offset) of the circle
% Note: the same result may be obtained if the following exchange is
% performed before the function call: q = -mu; mu = [0 0]';
E = [1 0;0 1]; % quadratic multiplier (unnecessary here; set to I)
tol = 1e-9;
p = mvnlps( mu, V, q, E, R, tol );
end

% ************************************************************************
% function mvnval = mvnlps( mu, sigma, q, e, r, re )
% MVNLPS Multivariate Normal Distribution Value for an ellipsoid.
% MVNVAL = MVNLPS( MU, SIGMA, Q, E, R, RE ) computes
% an MVN value to relative accuracy RE for an ellipsoid centered
% at Q with radius R and positive semi-definite ellipsoid matrix E:
% MVNVAL = PROB( ( X - Q )'E ( X - Q ) < R^2 )
% SIGMA is a positive definite covariance matrix for a multivariate
% normal (MVN) density with mean MU. MU and Q must be column vectors.
% Example:
% sg = [ 3 2 1;2 2 1;1 1 1]; mu = [1 -1 0]';
% e = [4 1 -1; 1 2 1; -1 1 2]; q = [2 3 -2]';
% p = mvnlps( mu, sg, q, e, 4, 1e-5 ); disp(p)
% Reference for basic algorithm is (S&O):
% Sheil, J. and O'Muircheartaigh, I. (1977), Algorithm AS 106:
% The Distribution of Non-negative Quadratic Forms in Normal Variables, Applied Statistics 26, pp. 92-98.
% Matlab implementation by
% Alan Genz, WSU Math, Pullman, WA 99164-3113; alangenz@wsu.edu.
% Listed in Calculate_protection_levels.m by permission 5/16/2009
% Transformation to problem with diagonal covariance matrix
% [ n, n ] = size(sigma); L = chol( sigma ); [ V D ] = eig( L*e*L' );
% cov = diag(D); d = sqrt(D)*(( inv(L)*V )'**( q - mu ));
% Basic S&O algorithm follows
% kmx = 2000; lambda = 0; covmx = max( cov );
% np = 0; rsqrd = r*r; A = 1;
% for j = 1 : n
%     if cov(j) > 1e-10*covmx
%         np = np + 1; gam(np) = 1;
%         alpha(np) = cov(j); A = A/alpha(np);
%         bsqrd(np) = d(j)^2/alpha(np);
%         lambda = lambda + bsqrd(np);
%     end
else
    rsqrd = rsqrd - d(j)^2;
end
end
if ( rsqrd <= 0 )
    mvnval = 0;
else
    covmn = min( alpha(1:np) );
    bet = 29*covmn/32; tbeta = rsqrd/bet; A = sqrt(A*bet^np);
    for j = 1 : np
        betalph(j) = bet/alpha(j);
    end
    c(l), c(2), ..., are S&O's c_0, c_1, ...;
    bsqrdf(j) is S&O's b_j^2 ; tbeta is S&O's t/beta, np is S&O's n';
    g(j) is S&O's g_j; gam(j) is S&O's gamma_j^k.
    csum = A*exp( -lambda/2 ); c(1) = csum; lgb = log( tbeta );
    if mod( np, 2 ) == 1
        F = erf( sqrt(tbeta/2) ); lgf = - tbeta/2 - log(2*pi*tbeta)/2;
        for nc = 3:2:np, lgf = lgb + lgf - log(nc-2); F = F - 2*exp(lgf); end
    else
        F = 1 - exp( -tbeta/2 ); lgf = - tbeta/2 - log(2);
        for nc = 4:2:np, lgf = lgb + lgf - log(nc-2); F = F - 2*exp(lgf); end
    end
    % equivalent F = gammainc( tbeta/2 , np/2 );
    mvnval = c(1)*F;
    for-loop computes S&O series
    for k = 1 : kmx
        gbsum = 0;
        for j = 1 : np
            gbsum = gbsum + k*bsqrdf(j)*gam(j)*betalph(j);
            gam(j) = gam(j)*( 1 - betalph(j) );
            gbsum = gbsum + gam(j);
        end
        g(k) = gbsum/2; cgsum = 0;
        for j = 0 : k - 1
            cgsum = cgsum + g(k-j)*c(j+1);
        end
        c(k+1) = cgsum/k; csum = csum + c(k+1);
        lgf = lgb + lgf - log(np+2*k-2); F = F - 2*exp(lgf);
        % equivalent F = gammainc( tbeta/2 , np/2 + k );
        mvnval = mvnval + c(k+1)*F;
        if ( 1 - csum )*F < re*mvnval, break; end
    end
end
% End function mvnlps
% compositePLdemo.m

function [VPL_H0,HPL_H0,RPL_H0] = compositePLdemo (v_or_h_flag,...
    PL_method,prob_level,pr_bias_mag,noise_scale_factor,SVs_to_remove)

% compositePLdemo: Test composite protection levels for GPS errors
%
% function [VPL_H0,HPL_H0,RPL_H0] = compositePLdemo (v_or_h_flag,...
%    PL_method,prob_level,pr_bias_mag,noise_scale_factor,SVs_to_remove)
%
% Parameters:
% v_or_h_flag = select dimension to process 1 = vertical, 2 = horizontal
% PL_method = method of computing protection level:
%   1   LAAS; use with zero bias for correct results
%   2   VPLc approximation 1, HPLc approximation 1 (overbounds)
%   3   VPLc approximation 1, HPLc approximation 2 (overbounds)
%   4   Exact VPLc and HPLc
%   5   RPLc approximation 1 (not yet assured to be an overbound)
%   6   Exact RPLc
% prob_level = level of probability associated with VPL and HPL, [0,1]
% pr_bias_mag = magnitude of uniform range domain biases on each SV
%   (Note: non-uniform biases may be obtained by entering -1)
% SVs_to_remove = row vector of SVs to remove from solution e.g., [3 22]
%
% VPL_H0 = vertical protection level (+/-) for the fault-free hypothesis
% HPL_H0 = horizontal protection level (+/-) for the fault-free hypothesis
% RPL_H0 = radial protection level for the fault-free hypothesis
%
% The demo is a test shell for the file Calculate_protection_levels.m. It
% uses a single simulated GPS measurement epoch using the 24-satellite
% Martinez constellation. It may be invoked with or without parameters.
% In the latter case, input for each parameter is requested from the user.
% This simulation runs out of steam on the typical Windows processor for
% number of data points exceeding 100,000. Therefore, it is most
% useful for illustrating the different protection level methods and
% verifying that they are correct.
%
% +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
% + Notice:   This software is intended for engineering development only +
% + and is not intended for use in applications involving safety of life +
% + or property. No warranties, expressed or implied, are made regarding +
% + the accuracy or correctness of this software or explanatory comments. +
% + User assumes all risks associated with use of this software. +
% +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
% (c) 2008-2009, Ohio University and/or Dean C. Bruckner, All rights reserved
% 2008-06-03 DCB Initial version, adapted from recursive version of LAAS
% MASPS Appendix J algorithm with biases added by Frank van
% Graas
% 2008-06-05 DCB Revised & added plots, corrected kappa def and std dev
% formula for horiz error; validated with FvG
% 2008-06-06 DCB Added conservative method for determining HPL_H0 as
% suggested by FvG (force all contributions to HPL to be
% collinear),
% 2008-07-31 DCB Added comments and verified algorithms again; expanded
user options.
% 2008-08-01 DCB Streamlined for just the original method for algorithm
% clarity.
% 2008-08-12 DCB Changed HPL algorithm to be consistent with VPL algorithm
% (i.e., multiply PR bias by S rows for East and North
% separately and then take root-sum square, instead of
% taking root-sum-square of East and North components of S
% and then multiplying by absolute value of bias)
% 2008-08-14 DCB Updated comments and plot labels
% 2008-08-21 DCB Implemented integration of ellipse/ellipsoid method
% instead of approximations using multiples of standard
% deviations as distance measures.
% 2008-08-29 DCB Vectorized some calculations for faster execution speed &
% implemented sequential VPL then HPL using a single bias
% combination, and HPL then VPL using a generally different
% bias combination. Revised & expanded simulation, printouts
% and plots.
% 2009-03-28 DCB Removed faulty algorithms ecda$c2 and ecda$c1 and replaced
% them with ecda$c3 and ecda$c3.
% 2009-05-16 DCB Added zero mean LAAS VPL and HPL for comparison
% 2009-05-17 DCB Put all VPL & HPL calcs into calculate_protection_levels.m
% 2009-05-19 DCB Completed verifying all cases.
% 2009-05-22 DCB Final dissertation version

% ============ INITIALIZE PARAMETERS ===============================

close all
time = 0;
lat = 30; % New Orleans
lon = -90;
svids =
  3
  4
  8
 13
 16
 19
 22];
azi =
   39.7271
  42.0439
 112.8938
 205.1581
 175.8236
 340.8624
 292.6822];
elv =
   48.5138
  14.1342
  36.8050
  32.2595
  8.9723
  74.0524
  30.4465];
Hmat =
   0.4234   0.5095   0.7491   1.0000
   0.6494   0.7201   0.2442   1.0000
   0.7376  -0.3115   0.5991   1.0000
  -0.3595  -0.7654   0.5338   1.0000
% From GAD-C1 ground/AAD-B air curves with multipath, iono, tropo:
%  gnd_acc_index = 1; % GAD-A; no specialized reference antenna, worst gnd noise/mp
%  air_acc_index = 2; % AAD-B, Best air noise
%  air_mp_index  = 1; % AAD-A; B is under development
%  num_ref_stas = 1; % Note that we will use K_ffmd for 2 ref stas (min # specified)
%  [err_rms,gnd_rms,air_rms,air_mp_rms,tropo_rms,iono_rms] = ...
% accdesf1(gnd_acc_index,air_acc_index,air_mp_index,num_ref_stas,theta_deg);
% For comparison, LAAS GAD-C1 ground/AAD-B air curves w/o MP, iono, tropo
% yield the following:
% sigma_total = [  
%  0.2398  
%  0.3001  
%  0.2720  
%  0.2822  
%  0.3246  
%  0.2224  
%  0.2824  
% ];
% disp('Uniform noise selected (for troubleshooting)')
% gnd_rms = ones(numvis,1)*.2;
% air_rms = ones(numvis,1)*.2;

numvis = length(svids);

0.0719  -0.9851  0.1560  1.0000
-0.0901  0.2596  0.9615  1.0000
-0.7954  0.3324  0.5067  1.0000];
% air_mp_rms = ones(numvis,1);

% ----------------- OBTAIN INPUTS IF NOT PROVIDED ---------------------

if nargin < 6
    disp(' ')
    disp(['SVs available: ',num2str(svids')])
    SVs_to_remove = input('Enter row vector of SVs to remove (e.g., [6 18])
(Enter = [22]): ');
end
if isempty(SVs_to_remove)
    SVs_to_remove = [22];
end

if nargin < 5
    disp(' ');
    noise_scale_factor = input('Enter scale factor to reduce noise sigma except iono & tropo: (<Enter> = 1): ');
end
if isempty(noise_scale_factor)
    noise_scale_factor = 1;
end
if noise_scale_factor < 0
    noise_scale_factor = abs(noise_scale_factor);
end

if nargin < 4
    disp(' ')
    pr_bias_mag = input('Enter uniform PR bias magnitude for each SV in meters (-1=non-uniform)
(<Enter> = 0.2; ): ');
end
if isempty(pr_bias_mag)
    pr_bias_mag = 0.2;
end
if pr_bias_mag ~= -1 % -1 is a flag to use non-uniform bias values
    pr_bias_mag = abs(pr_bias_mag);
end

if nargin < 3
    disp(' ')
    prob_level = input('Enter probability that must be contained within protection level:
(<Enter> = 0.95): ');
end
if isempty(prob_level)
    prob_level = 0.95;
end
if prob_level < 0 || prob_level > 1
    error('Probability value must be in interval [0,1]')
end

if nargin < 2
disp(' ')  
    PL_method = input(...  
    'Enter PL method (<Enter> = 4): ');  
end  
if isempty(PL_method)  
    PL_method = 4;  
end  

if nargin < 1  
    disp(' ')  
    v_or_h_flag = input(...  
    'Choose vertical (1) or horizontal (2) simulation mode (<Enter> = 1): ');  
end  
if isempty(v_or_h_flag)  
    v_or_h_flag = 1;  
end  
if v_or_h_flag == 2  
    if PL_method < 5  
        disp('Horizontal simulation mode selected')  
    else  
        disp('Radial simulation mode selected')  
    end  
else  
    v_or_h_flag = 1;  
    disp('Vertical simulation mode selected')  
end  

% Set bias bound vectors  
if pr_bias_mag == 0  
    disp('Bias value is zero for all satellites')  
    pr_bias_vector = zeros(numvis,1);  
    mu_ref = pr_bias_vector;  
    mu_air = pr_bias_vector;  
elseif pr_bias_mag == -1  
    disp('Using stored non-uniform biases')  
    mu_ref = 0.7*[  
      0.2103  
      0.1766  
      0.1888  
      0.1793  
      0.1799  
      0.2162  
      0.2000];  
    mu_air = 0.8*[  
      0.2457  
      0.1985  
      0.2300  
      0.1642  
      0.1922  
      0.2416  
      0.2000];  
else  

disp(['Using uniform bias value of ',num2str(pr_bias_mag),' m on all satellites'])
pr_bias_vector = pr_bias_mag*ones(numvis,1);
mu_ref = pr_bias_vector; % for testing only
mu_air = pr_bias_vector;
end
mu_total = hypot(mu_ref,mu_air);

% Simulate CNMP by reducing bias a little and noise/MP a lot
if noise_scale_factor > 0
    disp(['Scaling sigma (except iono & tropo) by factor of 1/',num2str(noise_scale_factor)])
gnd_rms = gnd_rms / noise_scale_factor;
air_rms = air_rms / noise_scale_factor;
air_mp_rms = air_mp_rms / noise_scale_factor;
end

% Compute sigma_total to be used in generating simulation error data
sigma_total = sqrt(gnd_rms.^2 + air_rms.^2 + air_mp_rms.^2 + tropo_rms.^2 + iono_rms.^2);
sig_ref = gnd_rms;
sig_air(:,1) = sqrt(air_rms.^2 + air_mp_rms.^2 + tropo_rms.^2 + iono_rms.^2);
% rms combination; these will be combined into sigma_total in Calculate_protection_level
% in the same way.
theta_deg = elv;

for k=1:length(SVs_to_remove)
    svr_index = find(svids == SVs_to_remove(k));
    if ~isempty(svr_index)
        svr_remvd = svids(svr_index);
        svids(svr_index) = [];
        elv(svr_index) = [];
        azi(svr_index) = [];
        sigma_total(svr_index) = [];
        Hmat(svr_index,:) = [];
        sig_air(svr_index) = [];
        sig_ref(svr_index) = [];
        mu_ref(svr_index) = [];
        mu_air(svr_index) = [];
        mu_total(svr_index) = [];
        theta_deg(svr_index) = [];
    end
numvis = length(svids);
disp(['SV ',num2str(svr_remvd), ' removed from simulation'])
end

if (numvis < 4)
    error('Number of SVs must be 4 or greater')
end

% =========== ESTIMATE VPL and HPL ================

K_ffmd = -sqrt(2)*erfcinv( 2 * ( 1 - ( 1 - prob_level ) / 2 ) )
P_hmi = 1 - prob_level;
% LS_weighting_type = 0; % = unweighted, 1 = wt by 1/bias, 2 = wt by 1/var
% This value is ignored for PL_method = 1 (LAAS) and wt by 1/var is used
if PL_method == 1
  LS_weighting_type = 2; % Use when bias << noise
  disp('Using LS weighted by 1/variance')
else
  LS_weighting_type = 0; % Use when bias & noise are comparable
  disp('Using unweighted LS')
  % LS_weighting_type = 1; % use when bias >> noise
  % disp('Using LS weighted by 1/bias')
end

% Find results of PL_method selected for comparison
[VPL_H0,HPL_H0,RPL_H0,C,S,D,bias_signs,H_proj_vector,biases_east,biases_north] = ...
  Calculate_protection_levels
  (Hmat,svids,numvis,mu_ref,sig_ref,mu_air,sig_air,...
   theta_deg,K_ffmd,P_hmi,PL_method,LS_weighting_type);
HPL_H0_sigma_east = sqrt(C(1,1));
HPL_H0_sigma_north = sqrt(C(2,2));
VPL_H0_sigma = sqrt(C(3,3));
RPL_H0_sigma_east  = HPL_H0_sigma_east;
RPL_H0_sigma_north = HPL_H0_sigma_north;

% Locate posn domain bias combination that would produce exact HPLc or
% RPLc, and use that particular bias combination to see if the PL
% algorithm works correctly. All the algorithms except LAAS recognize the
% biases provided as bounds, not actual biases, so the PLs produced by
% approximations should always be an overbound for exact methods 4 and 5.
if v_or_h_flag == 1 % vertical simulation mode
  bias_signs = sign(S(3,:)); % Worst case signs for vertical
  Biasxyzt(1:4) = S * (mu_total .* bias_signs') ;
  VPL_H0_bias = Biasxyzt(3); % for now use this for verification of routines
elseif v_or_h_flag == 2 % horizontal simulation mode
  if PL_method == 4 || PL_method == 5 || PL_method == 6
    % search already accomplished; bias_signs contain results
    Biasxyzt(1:4) = S * (mu_total .* bias_signs') ;
    % see Calculate_protection_level for computation of bias_signs
  else % Do minimal search to show worst-case bias for HPL in sim results
    % Create matrix of all possible sign combinations of uniform PR
    % bias for a given number of SVs
    ncom = 2^numvis; % number of signed bias combinations (+ or -)
    bias_combns = ones(ncom,numvis);
    for k1 = 1:numvis % number of columns
      ncyc = 2^k1; % number of cycles of +/- signs in this column
      for k2 = 1:ncyc
        bias_combns( ...
          (k2-1)*ncom/ncyc + 1 : k2*ncom/ncyc , k1 ) = ...
        ones( ncom/ncyc , 1 ) * (-1)^(k2+1);
      end
eend
    biases_east = bias_combns * (S(1,:)'.* mu_total);
    biases_north = bias_combns * (S(2,:)'.* mu_total);
    H_biases_projected = [biases_east biases_north] * H_proj_vector;
    [maxhb,ii] = max(abs(H_biases_projected));
    bias_signs = bias_combns(ii,:);
    Biasxyzt(1:4) = S * (mu_total .* bias_signs') ; % For RPL testing
  end
end
else
    error('v_or_h_flag must be 1 or 2')
end

if PL_method < 5
    HPL_H0_bias_east = Biasxyzt(1);
    HPL_H0_bias_north = Biasxyzt(2);
else
    RPL_H0_bias_east = Biasxyzt(1);
    RPL_H0_bias_north = Biasxyzt(2);
end

% ============ PERFORM SIMULATION TO COMPARE TO VPL AND HPL================

% Create simulated data & find horizontal and vertical ellipse statistics
num_iters = 1e5;
Noisexyzt = zeros(4,num_iters);
pct=100/num_iters;
disp('Creating simulation data; please wait...')
for k = 1:num_iters
    noisels=randn(numvis,1).*sigma_total;
    Noisexyzt(1:4,k) = S*noisels; % 4-D error vector (x,y,z and t);
end

if PL_method < 5
    % Univariate measures of vertical error
    % (i.e., projected on vertical axis of locally level coordinate frame
    % with bias if present; glideslope does not enter into this VPL version)
    V_err = Noisexyzt(3,:) + Biasxyzt(3);
    Vsigma = std(V_err); % 1-sigma value
    Vmean = mean(V_err); % 1-sigma value
    Vsort = sort(abs(V_err)); % for display of CDF
    pct_errors_within_VPL= (num_iters - ...
        ( length(find(V_err>VPL_H0)) + length(find(V_err<-VPL_H0)) ) ...
    )/num_iters; % 2-sided limit
    % Univariate measures of horizontal error
    % (i.e., projected onto noise ellipse major axis with bias if present)
    for k = 1:num_iters
        H_err(k) = dot(H_proj_vector,...
            [Noisexyzt(1,k)+Biasxyzt(1) Noisexyzt(2,k)+Biasxyzt(2)]);
        % error projected onto semi-major axis; is 2-sided (pos and neg)
    end
    Hsigma = std(H_err); % 1-sigma value
    Hmean = mean(H_err); % 1-sigma value
    Hsort = sort(abs(H_err)); % for display of CDF
    pct_errors_within_HPL= (num_iters - ...
        ( length(find(H_err>HPL_H0)) + length(find(H_err<-HPL_H0)) ) ...
    )/num_iters; % 2-sided limit
else
    % Univariate measures of vertical error
    % (i.e., projected on vertical axis of locally level coordinate frame
    % with bias if present; glideslope does not enter into this VPL version)
    V_err = Noisexyzt(3,:) + Biasxyzt(3);
    Vsigma = std(V_err); % 1-sigma value
    Vmean = mean(V_err); % 1-sigma value
    Vsort = sort(abs(V_err)); % for display of CDF
    pct_errors_within_VPL= (num_iters - ...
        ( length(find(V_err>VPL_H0)) + length(find(V_err<-VPL_H0)) ) ...
    )/num_iters; % 2-sided limit
    % Univariate measures of horizontal error
    % (i.e., projected onto noise ellipse major axis with bias if present)
    for k = 1:num_iters
        H_err(k) = dot(H_proj_vector,...
            [Noisexyzt(1,k)+Biasxyzt(1) Noisexyzt(2,k)+Biasxyzt(2)]);
        % error projected onto semi-major axis; is 2-sided (pos and neg)
    end
    Hsigma = std(H_err); % 1-sigma value
    Hmean = mean(H_err); % 1-sigma value
    Hsort = sort(abs(H_err)); % for display of CDF
    pct_errors_within_HPL= (num_iters - ...
        ( length(find(H_err>HPL_H0)) + length(find(H_err<-HPL_H0)) ) ...
    )/num_iters; % 2-sided limit
else
    % Univariate measures of vertical error
    % (i.e., projected on vertical axis of locally level coordinate frame
    % with bias if present; glideslope does not enter into this VPL version)
    V_err = Noisexyzt(3,:) + Biasxyzt(3);
    Vsigma = std(V_err); % 1-sigma value
    Vmean = mean(V_err); % 1-sigma value
    Vsort = sort(abs(V_err)); % for display of CDF
    pct_errors_within_VPL= (num_iters - ...
        ( length(find(V_err>VPL_H0)) + length(find(V_err<-VPL_H0)) ) ...
    )/num_iters; % 2-sided limit
    % Univariate measures of horizontal error
    % (i.e., projected onto noise ellipse major axis with bias if present)
    for k = 1:num_iters
        H_err(k) = dot(H_proj_vector,...
            [Noisexyzt(1,k)+Biasxyzt(1) Noisexyzt(2,k)+Biasxyzt(2)]);
        % error projected onto semi-major axis; is 2-sided (pos and neg)
    end
    Hsigma = std(H_err); % 1-sigma value
    Hmean = mean(H_err); % 1-sigma value
    Hsort = sort(abs(H_err)); % for display of CDF
    pct_errors_within_HPL= (num_iters - ...
        ( length(find(H_err>HPL_H0)) + length(find(H_err<-HPL_H0)) ) ...
    )/num_iters; % 2-sided limit
end
R_err = hypot( Noisexyzt(1,:), Biasxyzt(1) , Noisexyzt(2,:), Biasxyzt(2) );
Rsigma = std(R_err); %
Rmean = mean(R_err);
Rsort = sort(abs(R_err)); % for display of CDF
pct_errors_within_RPL = length(find(R_err < RPL_H0)) / num_iters; %

end

fprintf('
--- COMPARE VPL, HPL and RPL TO SIMULATION RESULTS ---------------------
');
fprintf('Simulation Results for Method %d
',PL_method)
fprintf('
Satellites: %d %d %d %d %d %d %d %d
',svids)

if v_or_h_flag == 1
    fprintf('
VPL_H0_sigma = %8.3f m 
VPL_H0_bias = %8.3f m
',VPL_H0_sigma,VPL_H0_bias)
else
    fprintf('
RPL_H0_sigma_east = %8.3f m 
RPL_H0_bias_east = %8.3f m
',RPL_H0_sigma_east,RPL_H0_bias_east)
    fprintf('
RPL_H0_sigma_north = %8.3f m 
RPL_H0_bias_north = %8.3f m
',RPL_H0_sigma_north,RPL_H0_bias_north)
end

if v_or_h_flag == 1
    fprintf('
VPL: %8.3f m, %8.4f%% (contains %8.4f%% of simulated vertical errors)
',VPL_H0,prob_level*100,pct_errors_within_VPL*100)
else
    fprintf('
HPL: %8.3f m, %8.4f%% (contains %8.4f%% of simulated horizontal errors)
',HPL_H0,prob_level*100,pct_errors_within_HPL*100)
end
fprintf('

RPL: %8.3f m, %8.4f%% (contains %8.4f%% of simulated radial errors)
',
RPL_H0,prob_level*100,pct_errors_within_RPL*100)
end
end

% =============================== PLOT RESULTS ============================
do_plot_flag = 1;
if do_plot_flag == 1;
    % Plot results
    if Pl_method < 5
        axis_p = round(max([HPL_H0 VPL_H0])) + 1.5; % m
    else
        axis_p = round(RPL_H0) + 1.5;
    end
    x1=cumsum(S(1,:)' .* (bias_signs' .* mu_total));x1=[0;x1];
y1=cumsum(S(2,:)' .* (bias_signs' .* mu_total));y1=[0;y1];
    vert_signs = sign(S(3,:));
z1=cumsum(S(3,:)' .* (vert_signs' .* mu_total));z1=[0;z1];
    len1 = sum ( hypot(S(1,:),S(2,:)) ) * .2;
    len2 = hypot(S(1,:)*mu_total,S(2,:)*mu_total);
    len3 = abs(S(3,:))*mu_total;

    figure
    subplot(311)
    plot(Noisexyzt(1,:)+Biasxyzt(1))
    axis([0 num_iters -axis_p axis_p])
    grid
    subplot(312)
    plot(Noisexyzt(2,:)+Biasxyzt(2))
    axis([0 num_iters -axis_p axis_p])
    grid
    subplot(313)
    plot(Noisexyzt(3,:)+Biasxyzt(3))
    axis([0 num_iters -axis_p axis_p])
    grid
    subplot(311)
    % title('Noise plus bias in East, North and Up directions, meters')
    if v_or_h_flag == 1 % Vertical
        lenv = 10;
        plot(Noisexyzt(1,:)+Biasxyzt(1),Noisexyzt(3,:)+Biasxyzt(3),'.','Color',[.6 .9 .5]),hold on
        plot(zeros(1,num_iters),Noisexyzt(3,:)+Biasxyzt(3),'r.'),hold on
        plot([-lenv,lenv],[VPL_H0,VPL_H0],'k-',[lenv,-lenv],[VPL_H0,]
        VPL_H0],'k-','Linewidth',2),hold on
        plot([0 Biasxyzt(1)],0 Biasxyzt(3),'b'),hold on
        plot(Biasxyzt(1),Biasxyzt(3),'bo','Linewidth',2)
        % text(-.9*axis_p,-.75*axis_p,['VPLc = ',num2str(VPL_H0),' m,'
        ','. .
        % num2str(prob_level*100),'
    end
end
% text(-.9*axis_p,-.9*axis_p, ['VPLc contains 
',num2str(pct_errors_within_VPL*100,... 
'\% of simulated vert posn errors'])
xlabel('East, m'); ylabel('Up, m')
axis([-axis_p axis_p -axis_p axis_p]); axis('square')
grid
% title('Bias and noise projected onto Up axis')
add_plot_flag_str = lower(input('Plot additional VPLs? ( 0=no, 1=y, <Enter>=n ): ','s'));
if isempty(add_plot_flag_str)
    add_plot_flag_str = 'n';
end
add_plot_flag_str(2:end)=[];
k = 1;
while strcmp(add_plot_flag_str,'y')
    plotstrs = ['k--';'k-.'
    if k == 3; k = 1; end
    plot([-lenv,lenv],
    k = k + 1;
    add_plot_flag_str = lower(input('Plot additional VPLs? ( y, n, <Enter>=n ): ','s'));
    if isempty(add_plot_flag_str)
        add_plot_flag_str = 'n';
    end
    add_plot_flag_str(2:end)=[];
end
% while add_plot_flag_str == 1
figure(gcf)
hold off

else
    figure
    plot(biases_east,biases_north,'m.')
end

if PL_method < 5 % HPLc
    % dx_sq = C(1,1);
    % dy_sq = C(2,2);
    % dxy =  C(1,2);
    % p1xy = ( dx_sq + dy_sq );
    % p2xy = ( dx_sq - dy_sq );
    % p3xy = ( dxy );
    % % Obtain errors projected onto major axis of horiz noise ellipse
    % thetal= 0.5*atan2(2*p3xy,p2xy);
    % thetal = thetal * 180/pi;
    % H_proj_vector = [cos(thetal) sin(thetal)];
    % CR = [ cos(thetal) -sin(thetal); sin(thetal) cos(thetal)]; % CCW rotation matrix
\[ \mathbf{CR} = [\mathbf{H}_{\text{proj vector}}(1) - \mathbf{H}_{\text{proj vector}}(2); \mathbf{H}_{\text{proj vector}}(2)] \]
\[ \mathbf{NB} = [\text{Noisexyzt}(1,:) + \text{Biasxyzt}(1); \text{Noisexyzt}(2,:) + \text{Biasxyzt}(2)] \]
\[ \mathbf{NB}_{\text{rotated}} = \mathbf{NB}' \times \mathbf{CR}; \text{ % Noise and bias rotated} \]
\[ \mathbf{NB}_{\text{projected}} = [\mathbf{NB}_{\text{rotated}}(:,1) \text{ zeros(length(Noisexyzt(1,:)),1)]}; \]
\[ \mathbf{NB}_{\text{rotated back}} = ([\mathbf{NB}_{\text{projected}}' \times \mathbf{CR}')'; \]
\[ \text{endpt1} = [-\text{HPL}_0 0] \times \mathbf{CR}'; \text{ % endpoints of HPL interval} \]
\[ \text{endpt2} = [\text{HPL}_0 0] \times \mathbf{CR}'; \]
\[ \text{m} = -\mathbf{H}_{\text{proj vector}}(1)/\mathbf{H}_{\text{proj vector}}(2); \text{ % slope of line perp to major axis} \]
\[ \text{lenh} = 10; \]
\[ \text{hash1x} = [\text{endpt1}(1)-\text{lenh} \text{ endpt1}(1)+\text{lenh}]; \]
\[ \text{hash1y} = [\text{endpt1}(2)-\text{m} \times \text{lenh} \text{ endpt1}(2)+\text{m} \times \text{lenh}]; \]
\[ \text{hash2x} = [\text{endpt2}(1)-\text{lenh} \text{ endpt2}(1)+\text{lenh}]; \]
\[ \text{hash2y} = [\text{endpt2}(2)-\text{m} \times \text{lenh} \text{ endpt2}(2)+\text{m} \times \text{lenh}]; \]
\[ \text{figure} \]
\[ \text{plot(Noisexyzt(1,:)+Biasxyzt(1),Noisexyzt(2,:)+Biasxyzt(2),'.','Color',[0 0.8 0.9]),hold on} \]
\[ \text{plot([-\mathbf{H}_{\text{proj vector}}(1) \mathbf{H}_{\text{proj vector}}(1)] \times \text{lenh},[-\mathbf{H}_{\text{proj vector}}(2) \mathbf{H}_{\text{proj vector}}(2)] \times \text{lenh},'k','Linewidth',1)} \]
\[ \text{plot(NB}_{\text{rotated back}(1,:),NB}_{\text{rotated back}(2,:),'r.'} \]
\[ \text{plot([0 \text{ Biasxyzt}(1)]',[0 \text{ Biasxyzt}(2)],'y-') \]
\[ \text{text(-axis_p*.1,-axis_p*.8, ['HPLc = ',num2str(HPL_0),', 'num2str(prob_level*100),', '+%'])} \]
\[ \text{text(-axis_p*.1,-axis_p*.9, ['HPLc contains ',num2str(pct_errors_within_HPL*100,...}} \]
\[ \%7.3f'), '% of simulated horiz posn errors'])} \]
\[ \text{title('Noise plus position domain biases that produced HPL')} \]
\[ \text{xlabel('East, m');ylabel('North, m')} \]
\[ \text{axis([-axis_p axis_p -axis_p axis_p]/2); axis('square')} \]
\[ \text{grid} \]
\[ \text{add_plot_flag_str = lower(input('Plot additional HPLs? ( y, n, <Enter>=n ): ','s'));} \]
\[ \text{if isempty(add_plot_flag_str)} \]
\[ \text{add_plot_flag_str = 'n';} \]
\[ \text{k = 1;} \]
\[ \text{while strcmp(add_plot_flag_str,'y')} \]
\[ \text{plotstrs = ['k--';'k-.'];} \]
\[ \text{plotstrs(k,:) = input('Enter additional HPL value for plotting (m): ', 's');} \]
\[ \text{HPL_temp = input('Enter additional HPL value for plotting (m): ', 's');} \]
\[ \text{plot(HPL_temp 0)'*CR'; % endpoints of HPL interval} \]
\[ \text{endpt2a = [HPL_temp 0]'*CR';} \]
\[ \text{m = -H_{proj vector}(1)/H_{proj vector}(2); % slope of line perp to major axis} \]
\[ \text{lenh} = 10; \]
\[ \text{hash1xa = [endpt1a(1)-lenh endpt1a(1)+lenh];} \]
\[ \text{hash1ya = [endpt1a(2)-m*lenh endpt1a(2)+m*lenh];} \]
\[ \text{hash2xa = [endpt2a(1)-lenh endpt2a(1)+lenh];} \]
\[ \text{hash2ya = [endpt2a(2)-m*lenh endpt2a(2)+m*lenh];} \]
\[ \text{plot(hash1xa,hash1ya,plotstrs(k,:),hash2xa,hash2ya,plotstrs(k,:),...} \]
    'Linewidth',2,'Color',[.5 .5 .5])
    k = k + 1;
    if k >= 3; k = 1; end
    add_plot_flag_str = lower(input('Plot additional HPLs? ( y, n <Enter>=n ): ','s'));
    if isempty(add_plot_flag_str)
        add_plot_flag_str = 'n';
    end
    add_plot_flag_str(2:end)=[];
end % while strcmp(add_plot_flag_str,'y')
figure(gcf)
hold off
else % RPLc

    figure

    plot(Noisexyzt(1,:)+Biasxyzt(1),Noisexyzt(2,:)+Biasxyzt(2),'.','Color',[0 0.8 0.9]),hold on
    plot(biases_east,biases_north,'y.'),hold on
    plot(x,y,'k-','Linewidth',2)
    plot([0 Biasxyzt(1)],[0 Biasxyzt(2)],'k-','Linewidth',2)
    text(-axis_p*.1,-axis_p*.8,['RPLc = ',num2str(RPL_H0),'
           m, ',...
           num2str(prob_level*100),'
           %'])
    text(-axis_p*.1,-axis_p*.9, ['RPLc contains
           ',num2str(pct_errors_within_RPL*100,...
           '%7.3f'),'
           % of simulated horiz posn errors'])
    title('Noise plus position domain biases that produced RPLc')
    xlabel('East, m');ylabel('North, m')
    axis([-axis_p axis_p -axis_p axis_p]); axis('square')
    grid
    add_plot_flag_str = lower(input('Plot additional RPLs? ( y, n, <Enter>=n ): ','s'));
    if isempty(add_plot_flag_str)
        add_plot_flag_str = 'n';
    end
    add_plot_flag_str(2:end)=[];
    k = 1;
    while strcmp(add_plot_flag_str,'y')
        plotstrs = ['k--';'k_.'];
        RPL_temp = input('Enter additional RPL value for plotting (m): '
           ',%RPLc temp:RPL_temp/100:RPL_temp; % Protection level circle
to be plotted
        ya=sqrt(RPL_temp^2 - xa.^2);
        plot(xa,ya,plotstrs(k,:),xa, y,plotstrs(k,:),',Linewidth',2,'Color',[.5 .5 .5]),hold on
        k = k + 1;
        if k >= 3; k = 1; end
        add_plot_flag_str = lower(input('Plot additional RPLs? ( y, n, <Enter>=n ): ','s'));
        if isempty(add_plot_flag_str)
            add_plot_flag_str = 'n';
        end
        add_plot_flag_str(2:end)=[];
end % while strcmp(add_plot_flag_str,'y')
hold off
figure(gcf)
end % PL_method < 4

end % if v_or_h_flag == 1

end % if do_plot_flag == 1

save