SAFETY CRITICAL BOUNDS FOR PRECISE POSITIONING FOR AVIATION AND AUTONOMY

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Kazuma Gunning
March 2021
I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Todd Walter, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Grace Gao

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

J Powell

Approved for the Stanford University Committee on Graduate Studies.

Stacey F. Bent, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.
Abstract

Unmanned aerial vehicle (UAV) and autonomous platforms can greatly benefit from an assured position solution with high integrity error bounds. The global navigation satellite system (GNSS) offers nearly ubiquitous positioning, and the expected high degree of connectivity in these vehicles will allow users to receive real time precise clock and ephemeris corrections to the GNSS navigation messages. Such corrections enable the use of precise point positioning (PPP) techniques. Up to now, these techniques have mostly been used to provide high accuracy, rather than focusing on high integrity applications. In this thesis, I apply the methodology and algorithms used in aviation to determine position error bounds with high integrity (or protection levels) for a PPP position solution. A navigation system that incorporates measurements from GNSS, an inertial measurement unit (IMU), and an odometer, and is tolerant to faulted measurements was developed and demonstrated with static, automobile, and flight data. Methods are developed and discussed that reduce the complexity and computational load of the system. In order to stress the error detection capabilities, faults are injected into the measurements. The overall position error bounds produced, which are often under two meters, offer a significant improvement over the state of the art in high integrity navigation, which is driven by the aviation industry and are on the order of tens of meters. Another contribution relaxes the requirements on the use of precise orbit and clock corrections by the navigation filter while still producing tighter error bounds through the use of precision navigation techniques. A final topic discussed involves techniques developed for measuring the errors affecting the GNSS satellites as they relate to GNSS constellation monitoring, which is required to support advanced receiver autonomous integrity monitoring (ARAIM).
ARAIM is a high integrity navigation technique in development for aviation and that was adapted here to provide the protection levels for PPP. Significant temporal variations in the satellite performance are observed in ways that are undetectable to other constellation monitoring techniques.
Acknowledgements

First, I must acknowledge Professor Per Enge. I think very few people enter their undergraduate careers wanting or expecting to study navigation, but the magnetic presence and teaching of Professor Enge made it the field I wanted to be in. He always pushed me to do better and to do more while always maintaining a smile and a great environment. I know that I miss him, and I know that everyone who had the fortune of interacting with him does too.

Professor Todd Walter has been guiding my career in some way for almost a decade, and for that I am very grateful. He has been there to steady me when I am panicking about a conference in a week, and he has been there to encourage me to do even more than I thought I could do. He also knows just an incredible amount about GNSS, which helps, too.

I am grateful that Professor J. David Powell served as my advisor during a crucial transition period. He always brought great insights and energy to our discussions, which I will always remember. The other members of my defense and reading committees were crucial in getting me finally over the hump of defending and submitting. Professor Simone D’Amico’s classes were among the most interesting and useful of my time at Stanford, and you have always been a very nice person to chat with while you get your espresso in the GPS lab. Professor Leo Hollberg served as my defense chair, and I know that I and the rest of the GPS lab are very appreciative of your willingness to participate in our defenses. Professor Grace Gao was actually the first person I collaborated with in the Stanford GPS lab not too long after she earned her PhD, and I am very happy that she came back to Stanford.

The Stanford GPS Lab is truly an incredible place to work, and it’s because of the
other people there. The open-door policy of the incredible staff fosters an atmosphere of collaboration and inspiration, and my work quite literally would not have happened without the help of those in the lab. Dr. Juan Blanch is not only a brilliant guy whose work was the foundation of a lot of what is in this thesis, but he also has the closest movie taste to mine in the lab, which I also appreciate. I deeply admire Dr. Sherman Lo for his commitment to helping others through advisory and teaching roles, and I appreciate his insights on my work from a slightly more outside perspective. Dr. Eric Phelts has helped me and my work to grow and reach new audiences. Dr Yu-Hsuan Chen truly is the bedrock of the lab, and very little would get done without him.

Beyond the staff, I am thankful for my labmates, who both enriched and probably lengthened my time as a PhD student. You have become some of my closest friends and I am beyond lucky to have started the next phase of my life with a few of you. So, thank you to Dr. Adrien Perkins, Dr. Andrew Neish, Dr. Tyler Reid, Fabian Rothmaier, and ShiWen Zhang.

My time during my PhD wasn’t just spent in the lab. We went on trips to Vegas and New Orleans, watched fights, griped about stuff over gchat, hung out among the redwoods, and just hung out in general. These last few years would have been a lot less bright without Emily Ashton, Angela Chu, Jason Gonzalez, Dan Ibarra, Erin Ladd, Brian Louie, Sarah Lummis, Zach Ming, Briana Okyere, Poncie Rutsch, and Minna Zhang. You all inspire me on a daily basis, and I’m lucky to be your friend.

My family always pushed me to do well in school. Little did they know, I’d be in school until I was 31 years old. Jokes aside, without your support and encouragement, I’m not sure I would have been brave enough to embark on this journey, and I am very thankful that I did. Mom, Dad, Bob, Dora, and Hannah- thank you. I would also like to thank my in-laws, Lana and Peter, who have welcomed me and put up with their son-in-law being in school until this advanced age.

Krystina- doing this PhD has not been the easiest thing at times, but you have been there with me every step of the way. This is for you. Thank you. I love you.
# Contents

Abstract v

Acknowledgements vii

1 Introduction 1

1.1 Motivation .................................................. 1

1.2 Contributions ................................................ 3

1.2.1 Precise Positioning with Integrity ...................... 3

1.2.2 PPP Integrity Techniques with Broadcast Navigation Messages 3

1.2.3 Constellation Monitoring ............................. 4

1.3 Outline ....................................................... 4

2 Background 7

2.1 Precise Point Positioning .................................. 8

2.1.1 Modeling of GNSS Observations ...................... 16

2.2 Tightly Coupled Inertial Measurement Unit ............ 27

2.3 Advanced Receiver Autonomous Integrity Monitoring (ARAIM) ... 30

2.3.1 Multiple Hypothesis Solution Separation .............. 31

2.3.2 System Architecture .................................... 33

2.3.3 Integrity Support Message .............................. 35

3 PPP with Integrity 37

3.1 Introduction ................................................... 37

3.2 Prior Art ...................................................... 38
3.3 Filter Description .................................................. 39
3.3.1 Subset Management ........................................... 40
3.3.2 State Management ............................................. 43
3.3.3 Time Update ..................................................... 44
3.3.4 Measurement Models .......................................... 44
3.3.5 Measurement Update with Residual Checks ................. 48
3.3.6 Inclusion of an IMU ........................................... 48
3.3.7 Computing Protection Levels ................................. 49

3.4 Results .............................................................. 58
3.4.1 Datasets ......................................................... 58
3.4.2 Protection Level Algorithm Selection ......................... 62
3.4.3 Static Open Sky Scenario ..................................... 64
3.4.4 Fault Injection Scenarios ...................................... 66
3.4.5 Dynamic Open Sky Scenario .................................. 68
3.4.6 Dynamic results in open sky conditions with two out .... 68
3.4.7 Suburban Scenario ............................................. 69
3.4.8 Highway Scenario ............................................... 71

3.5 Conclusion .......................................................... 72

4 Integrity with Broadcast Navigation Data .......................... 74
4.1 Introduction .......................................................... 74
4.1.1 Prior Art .......................................................... 76
4.2 Methods ............................................................. 76
4.2.1 Estimator Design ............................................... 76
4.2.2 Integrity Algorithm Design ..................................... 77
4.3 GPS Performance Characterization ............................... 79
4.3.1 Inclusion of WAAS Corrections ............................... 81
4.4 Results .............................................................. 83
4.4.1 Static Scenario .................................................. 83
4.4.2 GPS-Only Flight Data .......................................... 85
4.4.3 GPS-GAL Flight Data ........................................... 85
4.5 Fault Detection and Exclusion ........................................... 88
  4.5.1 Original method for FDE ............................................ 89
  4.5.2 New method for FDE with “on-deck” filters ......................... 91
4.6 Conclusion ..................................................................... 93

5 Constellation Monitoring ................................................. 94
  5.1 Introduction ................................................................. 94
  5.2 Prior Art ..................................................................... 97
  5.3 Clock Bias and DCB Estimation ......................................... 98
    5.3.1 Ground Receiver Selection ......................................... 98
    5.3.2 Measurement Modeling ............................................. 99
    5.3.3 Evaluating ARAIM Dual Frequency Ranging .................... 101
    5.3.4 Estimator Structure .................................................. 102
  5.4 Results ...................................................................... 105
    5.4.1 L1P-L2P Clock Estimation Results ................................. 105
    5.4.2 L1 C/A- L5Q Clock Estimation Results ............................ 106
    5.4.3 DCB Estimation Results ............................................. 109
  5.5 Conclusions .................................................................. 113

6 Conclusions ..................................................................... 114
  6.1 Summary and Conclusions .............................................. 114
    6.1.1 PPP with Integrity .................................................... 114
    6.1.2 Integrity with Broadcast Navigation Data ........................ 115
    6.1.3 Constellation Monitoring ............................................ 116
  6.2 Directions for Future Work ........................................... 117

A Constellation Monitoring ADD ........................................... 119
  A.1 Algorithm Objective ..................................................... 119
  A.2 Assumptions and Limitations ......................................... 121
  A.3 Analyses and Trades ..................................................... 121
  A.4 Monitor Inputs ............................................................. 122
    Monitor Inputs ................................................................. 122

# List of Tables

2.1 Example Process Noise ............................................. 13  
2.2 Key ISM Parameters ................................................. 36  

3.1 Initial State Uncertainties ........................................ 43  
3.2 Process Noise Settings ............................................ 45  
3.3 Measurement Noise Settings ...................................... 47  

A.1 Monitor Inputs ....................................................... 123  
A.2 Precise Orbit Ingestion Inputs ................................. 123  
A.3 Precise Clock Ingestion Inputs ............................... 125  
A.4 Receiver Logged Broadcast Navigation Message Ingestion Inputs ... 126  
A.5 Navigation Message Voting Inputs ............................. 127  
A.6 Navigation Message Voting Outputs ........................... 127  
A.7 Precise Orbit Interpolation Inputs ............................. 128  
A.8 Precise Orbit Interpolation Outputs ........................... 129  
A.9 Broadcast Navigation Message Propagation Inputs .......... 129  
A.10 Broadcast Navigation Message Propagation Outputs ....... 130  
A.11 Clock Synchronization Inputs .................................... 131  
A.12 Clock Synchronization Outputs .................................. 131  
A.13 Satellite Orbital Attitude Adjustment with Antenna Phase Center Offset Inputs ................................................. 132  
A.14 Satellite Orbital Attitude Adjustment with Antenna Phase Center Offset Outputs ................................................. 133  
A.15 Local Frame Conversion Inputs .................................. 133
List of Figures

2.1 Extended Kalman filter flow diagram ................................. 11
2.2 Static PPP Solution Example ........................................ 17
2.3 UNB3 zenith dry delay time variation ............................... 21
2.4 UNB3 zenith wet delay time variation ............................... 22
2.5 UNB3 zenith delay variation with user elevation .................... 23
2.6 Mapping of Niell ....................................................... 24
2.7 Radial displacement of Earth Solid Tides over a day .......... 26
2.8 Tightly Coupled IMU Structure ..................................... 28
2.9 ARAIM Offline Monitoring Architecture [1] ....................... 34
2.10 Constellation monitoring structure ................................ 36

3.1 PPP with Solution Separation Implementation Diagram .......... 40
3.2 Grouping fault modes in a single subset reduces total subset count 41
3.3 Effect of grouping on subset count .................................... 42
3.4 Receiver antenna used to collect the static data at Stanford University 58
3.5 Open sky dynamic data set collected in Calgary .................. 59
3.6 Dynamic data collected in a midtown suburban environment .... 60
3.7 Dynamic data collected in a highway environment ................ 61
3.8 Dynamic flight data .................................................. 61
3.9 Position error and protection levels for the static scenario- on the right is a zoom in ....................................................... 62
3.10 Position error and protection levels for the dynamic scenario- on the right is a zoom in ................................................... 63
3.11 Run time as a function of the number of subsets .................. 65
3.12 Position errors and PLs in static scenario- on the right is a zoom in  
3.13 PRN 8 Measurement Usage with Injected Step Fault ................. 66 
3.14 Protection Levels with Injected Step Fault ......................... 66 
3.15 Injected Ramp Fault in Static Scenario ................................. 67 
3.16 Position errors and PLs for dynamic scenario in open sky- on the right 
is a zoom in ........................................................................... 68 
3.17 Position errors and PL for dynamic scenario in open sky conditions 
with two satellites out- on the right is a zoom in ....................... 69 
3.18 Position errors and PLs for suburban dynamic scenario .......... 70 
3.19 Position errors and PLs for highway dynamic scenario .......... 72 

4.1 Errors in the GPS broadcast orbit and clock are correlated over time 78 
4.2 Change in SISRE per second ............................................... 80 
4.3 Staggered IODE Changeover ............................................... 81 
4.4 WAAS Broadcast Orbit and Clock Corrections ...................... 82 
4.5 Position error and protection levels using broadcast navigation data 84 
4.6 Position error and protection levels for GPS-only flight data ...... 86 
4.7 Ground track and satellite visibility for GPS-Galileo example flight data 86 
4.8 Horizontal (left) and vertical (right) position error and protection levels 
for GPS-Galileo example flight data comparing snapshot ARAIM and 
sequential filters using only broadcast navigation data ............... 87 
4.9 Summary of mean horizontal (left) and vertical (right) protection levels 
across 35 flights comparing snapshot ARAIM and sequential filters using only broadcast navigation data .................. 88 
4.10 Injected Ramp Fault in Static Scenario ................................. 89 
4.11 Measurement usage diagram for original fault detection and exclusion 
method ................................................................................ 90 
4.12 Position error and protection levels given injected ramp fault and original FDE method .............................................. 90 
4.13 Measurement usage diagram for "on-deck" fault detection and exclusion method ...................................................... 91
4.14 Position error and protection levels given injected ramp fault and "on-deck" FDE method 

5.1 GPS Constellation Performance Overview comparing broadcast navigation message to NGA precise orbit and clock: 2008-2018

5.2 Number of ground receivers in view at GPS altitude

5.3 Clock differences and differential code biases

5.4 GPS CNAV Values of L1 C/A - L5Q Dual Frequency DCB

5.5 Comparison of SU GPS vs IGS CODE clock bias estimation for 2018

5.6 SU Estimated L1 C/A - L5Q Difference from L1P-L2P

5.7 L1 C/A - L5Q Clock Difference from L1P-L2P

5.8 Histogram of L1 C/A - L5Q Clock Difference from L1P-L2P: SU-CNAV

5.9 L1 C/A-L5Q Daily Variation and Beta Angle for PRN 10

5.10 Daily variations in L1 C/A - L5Q vs L1P-L2P clock bias at high and low beta angles

5.11 Difference between Daily SU and DLR Differential Code Bias

5.12 Difference between High Rate SU and Other DCB

5.13 High Rate DCB Estimate for PRN 6 Under high C/A

5.14 Histogram of Difference between High Rate SU and CNAV DCB

A.1 MPE determination on the left, a grid of users in the middle, and a comparison of MPE and UPE on the right.

A.2 MPE (heavy black line) versus UPEs (multicolored lines) for a particular satellite over time

A.3 .1.51: Geometric method for computing Maximum Projected Error

A.4 .1.52: MPE (heavy black line) versus UPEs (multicolored lines) for a particular satellite over time with mean error removed

B.1 Vehicle Sensors

B.2 Dataset Location Satellite View

B.3 Ground Track of Vehicle Dataset

B.4 Gap in GNSS Observation Availability
B.5 Position Error and Protection Level for estimator with GNSS Only . 150
B.6 All in View and Subset Position Errors- GNSS Only . . . . . . . . . 151
B.7 Position Error and Protection Level for estimator with PPP + IMU +
   Constraints . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 152
B.8 Position Error and Protection Level for estimator with PPP + IMU +
   Constraints + Odometry . . . . . . . . . . . . . . . . . . . . . . . . . . . 154
B.9 Position Error and Protection Level for estimator with PPP + IMU +
   Constraints + Odometry with only 10 seconds GNSS initialization . . 155
B.10 Ground track for estimator with PPP + IMU + Constraints + Odom-
    etry with only 10 seconds GNSS initialization . . . . . . . . . . . . 156
B.11 All in View and Subset Position Errors for estimator with PPP + IMU
    + Constraints + Odometry with only 10 seconds GNSS initialization 157
Chapter 1

Introduction

1.1 Motivation

The global navigation satellite system, or GNSS, has changed the way that people navigate by providing a ubiquitous, all weather, and meter-level accurate service. The American GNSS constellation, the Global Positioning System (GPS) is the most well known of the core constellations, but the Russian GLONASS, European Galileo, and Chinese BeiDou have been developed to offer similar services. Each core constellation operates in a similar fashion. Satellites orbit in medium Earth orbit (MEO) approximately 20,000 kilometers above the Earth and broadcast ranging signals. A ground segment monitors those signals, estimates the orbits and clock states of the satellites, and periodically uploads navigation messages for the satellites to broadcast. The user segment receives the ranging signals and navigation messages in order to estimate its own position. Such receivers are cheap and plentiful, with the number of GNSS receivers globally reaching into the billions [3].

Applications of GNSS abound, ranging from providing timing to critical infrastructure to allowing for turn-by-turn navigation on a smart phone. For many applications, occasional erroneous position outputs from the receiver are acceptable. For high integrity, safety critical applications, however, there must be a bound to the errors on the position solution and some guarantee of performance. Integrity is the level of trust that can be placed on the navigation system. Because a single GNSS
solution is constructed from many measurements, error checking of the measurements against one another can be performed using a technique called receiver autonomous integrity monitoring (RAIM). The aviation industry has been using GNSS and RAIM in safety critical applications since the 1990s [6]. RAIM ultimately produces a protection level associated with the position solution, which is a bound on the position error to a specified probability. The bounds produced by RAIM and even its multi-constellation successor, Advanced RAIM (ARAIM) are on the order of tens of meters. As the desire for commercial autonomous applications grows, the requirements on the protection levels produced become more stringent, and the techniques from the aviation industry are no longer able to meet them.

The Society of Automotive Engineers describes different levels of autonomy, where level 1 vehicles have no autonomy or assistance features. This comprises most cars built through the 1970s. Many research and commercial groups are racing to develop level 4 autonomous vehicles. Level 4 vehicles may allow an option for the user to take control and drive, but the vehicle can operate fully autonomously. The navigation requirements on such a vehicle are very strict and require lane-level navigation and corresponding decimeter accuracy with high integrity [68]. Unmanned aerial vehicles (UAV) and urban air mobility with autonomy are also emerging fields with high performance demands [9]. Current GNSS techniques are insufficient to support the needs of these industries.

In this thesis, I developed techniques to support the rigorous demands of emerging autonomous vehicles by integrating the high integrity techniques developed for aviation with a high accuracy GNSS technique called precise point positioning (PPP) as well as a tightly couple inertial measurement unit (IMU). The precision GNSS and sensor fusion navigation filter allow for lower protection levels than the previous industry standards. The primary goals of this thesis are to improve performance of high integrity navigation techniques.
1.2 Contributions

My first contribution is creating a multi-sensor high integrity, high accuracy navigation system combining PPP, IMU, and odometry. The second contribution is using similar techniques to the first contribution to dramatically lower protection levels even for users with GNSS only and no external corrections. The third contribution is a constellation monitoring and clock estimation system that can observe previously transparent errors in the code and carrier of the signals used for aviation. The third contribution also supports the assumptions used in the first two contributions.

1.2.1 Precise Positioning with Integrity

I designed and prototyped a navigation system with high integrity, accuracy, and continuity using a tightly coupled PPP-IMU-Odometry system combined with high integrity algorithms originally developed for aviation. This system relies on banks of navigation filters that monitor various fault hypotheses. I enabled the use of the system by reducing computational complexity through reduction of the computational cost savings per filter and through the grouping of multiple fault hypotheses to lower the number of required filters. I improved continuity through the tight coupling of an IMU as well as integration of other sensors. Additionally, I developed a method by which “on-deck” subset filters are used to maintain low protection levels even through fault exclusion. Ultimately, the algorithms I developed produce meter-level protection levels that rigorously bound solution error. These protection levels are an order of magnitude lower than their aviation counterparts. The high integrity navigation system has been demonstrated with static, automobile, and flight data in a variety of environments both nominally and with injected faults.

1.2.2 PPP Integrity Techniques with Broadcast Navigation Messages

My second contribution builds off the first as I significantly reduced protection levels relative to ARAIM using only broadcast navigation messages by using carrier phase
measurements and exploiting the temporal correlation of the ephemeris error with a PPP-like filter and solution separation. I produced small protection levels using only information available from GPS broadcast and/or the Wide Area Augmentation System (WAAS) and banks of sequential filters. In order to do this, I analyzed broadcast navigation message error over time. The performance of this high integrity navigation system was evaluated over dozens of flights and hundreds of hours of data collection. The use of this new navigation filter drives down protection levels compared to snapshot estimators that are currently used in aviation.

1.2.3 Constellation Monitoring

My third contribution sought to validate the performance assumptions that go into ARAIM as well as the application of ARAIM to the previous contributions. The use of ARAIM requires the careful characterization of GNSS constellation performance. I developed methods to examine ranging errors using the coarse pseudorange and precise carrier phase signals to identity and characterize errors that were previously undetectable to other constellation monitoring techniques. I built a highly accurate estimator of GNSS satellite clock and bias parameters that are better than nanosecond accurate. I observed code biases that have impacted dual-frequency GNSS users up to one meter over all of the year 2018. The second part of this contribution was to develop, automate, and expand other constellation monitoring software that compares the broadcast navigation messages to post-processed precise orbit and clock products. I analyzed millions of satellite hours across four GNSS constellations.

1.3 Outline

Chapter 2 provides background information critical to the techniques used in this thesis. An overview of PPP is provided through a description of the technique as a whole and a walk through of a simple example of an implementation. A description of the various ranging error models used in PPP follows. Further background covers the inclusion of an IMU into the PPP algorithm. Chapter 2 closes with a description
1.3. OUTLINE

of ARAIM. An overview of the system, a description of the system architecture, and the methodology behind producing protection levels are presented.

Chapter 3 introduces my first contribution, which is to develop techniques that integrate the accuracy of PPP with the integrity of ARAIM. After examining the prior art in the field, a description of the filter walks through the subset and state management, the time and measurement update, and the generation of protection levels. Data sets from static open sky, open sky automobile, suburban automobile, highway automobile, and aircraft scenarios are examined. A protection level algorithm is selected from a set of candidate algorithms, and the fault detection capabilities are tested through a series of injected faults. The meter-level protection levels produced are validated against truth position data associated with the data sets.

Chapter 4 extends the techniques developed in Chapter 3 to use without the precise orbit and clock corrections that are typically required for PPP. Instead, broadcast GPS navigation messages are used with and without WAAS corrections. An analysis of GPS clock and ephemeris errors is performed, and protection levels are produced for dozens of flights. The protection levels from these newly developed techniques are compared to those produced by snapshot ARAIM, and it is shown that the protection levels produced by the work in this thesis out perform those from ARAIM.

Chapter 5 develops techniques that can be used directly in the ARAIM system by improving constellation monitoring techniques. Said techniques can be used to perform the statistical analysis of constellation performance required to produce ARAIM protection levels. In particular, this chapter develops an automated global satellite clock and bias monitoring system through the use of a network of GNSS receivers. A description of the network is followed by the development of the system that estimates precise GNSS clock states and biases that are important for the dual frequency ARAIM user. An analysis of a year of GPS data shows significant deviation from previously examined values for differential code biases due to daily effects.

Chapter 6 concludes the main portion of the thesis with a summary of my contributions as well as potential future work following this.

Appendix A describes the constellation monitoring algorithms that the results of Chapter 5 would feed into. Appendix B is an extension of the work done in Chapter
3 that includes the use of odometry measurements from an automobile into the PPP and protection level algorithms.
Chapter 2

Background

This chapter provides background information about several of the primary techniques used in the methodology developed in later chapters. Namely, precise point positioning (PPP), the tight coupling of an IMU and GNSS receiver, and advanced receiver autonomous integrity monitoring (ARAIM) are described. PPP is a GNSS positioning technique that can be used without any local base stations and offers decimeter-level root-mean-square (RMS) or better positioning through the use of dual frequency carrier phase measurements and precise orbit and clock estimates. IMU tight coupling allows for the inclusion of IMU measurements in the PPP solution while simultaneously estimating IMU error correction terms. Finally, ARAIM is a multi-constellation, multi-frequency GNSS integrity technique that, given redundant measurements, can produce bounds, often referred to as protection levels (PL), on the positioning error for safety critical applications. This chapter does not include an overview of the basics of GNSS, which can be found in [53].

Before proceeding, a few brief definitions of terms relating to navigation system performance will be offered [31]. Availability is the percentage of time that the navigation service is usable. Continuity is the probability that the service will be uninterrupted over a period. Frequent, very short outages may not affect availability significantly, but they will be detrimental to continuity. As previously mentioned, integrity is a measure of trust in the navigation system. The navigation system should be able to provide protection levels, which are bounds on the navigation solution
error to a specified probability. The probability that the solution error exceeds the 
protection level should be at or below the specified probability. The general approach 
to integrity that has been taken by the aviation community has been to prove integrity 
through the careful development of a threat model, the nominal distributions, and 
the probabilities of failure associated with it. Using that threat model, where each of 
the components is proven, a user navigation algorithm can be developed that meets 
the desired level of integrity and performance. This is in opposition to an approach 
where a technique is developed and proven out empirically through data collection. 
Due to the very small probabilities involved, such an approach is infeasible and would 
not adequately cover all feared events.

2.1 Precise Point Positioning

Precise point positioning is a GNSS estimation algorithm that relies on external orbit 
and clock corrections to produce highly accurate absolute navigation solutions using 
a single receiver [50]. PPP takes the approach of using high fidelity models of as 
many ranging error sources as possible and introducing estimated parameters related 
to those ranging errors when modeling is not possible. One such estimated parameter 
is the carrier phase ambiguity. Carrier phase measurements are very precise but 
contain an unknown ambiguity term. A PPP solution uses a sequential filter, such 
as an extended Kalman filter, to estimate those carrier phase ambiguities. These 
ambiguities start off with large uncertainty, but as the satellites move across the 
sky and the geometry changes, the uncertainty decreases and the position solution 
converges to decimeter or even centimeter levels of RMS error. Specifically, this 
thesis primarily deals with PPP using float carrier ambiguities, where the carrier 
phase ambiguities do not fix to integer values. The primary alternative to PPP is a 
technique called real-time kinematic (RTK), which seeks to eliminate ranging errors 
through double differencing measurements from a receiver at a base station of a known 
location and those from the rover receiver at the unknown location.

To illustrate the difference between the two techniques, we take the mitigation of 
orbit and clock error as an example. The difference between the true satellite position
2.1. PRECISE POINT POSITIONING

and clock state and the position and clock state broadcast in the navigation message can contribute tens of centimeters of RMS ranging error. The projection of the orbit and clock error onto the line of sight of two nearby receivers is nearly identical, and by differencing measurements from those two receivers, RTK can eliminate nearly all the error. PPP takes a very different approach, where a global network of receivers estimates the orbits and clocks at cm-level. That information, comprised of position and clock bias information, is provided to the receiver to use in its measurement prediction, which also eliminates the error from the broadcast navigation message.

There are many attractive elements of PPP, not the least of which is that it is a globally available solution because it does not require a nearby base station or network of base stations as differential techniques do. It can provide decimeter accurate solutions in dynamic scenarios and centimeter accurate static solutions. The high solution availability also makes PPP desirable, where as long as measurements are available, a float PPP solution can be formed, whereas solutions that rely on ambiguity fixing are often unavailable \[69\]. For a user that requires high integrity in a tough signal environment like an urban canyon, it may be difficult to rely on a solution that may or may not be available from epoch to epoch, such as RTK.

Downsides to PPP include the requirement of precise orbit and clock products. While there is no need for differential corrections from a nearby receiver, there is still a need for real-time orbit and clock products. Even with the corrections, which can include orbit, clock, and ionospheric products, convergence time can be tens of minutes to reach high accuracy. For single constellation solutions, convergence times can be thirty minutes or more. Finally, the accuracy of PPP is typically worse than short-baseline RTK solutions, which can be cm-level even in dynamic scenarios.

There are currently many commercial and free PPP services. Commercial services such as Novatel’s TerraStar [48], Trimble’s RTX [25], GMV’s magicGnss [80], and Swift’s Skylark offer real-time precise navigation solutions. All of these services provide orbit and clock corrections as well as ionospheric corrections for single-frequency users. Some of these also offer data streams from geostationary satellites to enable truly global solutions. Typically, these services require the use of a receiver produced by the company providing the service, and the correction data provided to the receiver
is not exposed to the user. In addition to commercial services, there are also several free PPP post-processing services. In this case, the user uploads GNSS measurement data in RINEX format to the server, and either a dynamic or static solution is returned. Two of the best free PPP services are Natural Resources Canada’s Canadian Spatial Reference System (CSRS) PPP tool [78] and NASA Jet Propulsion Laboratory’s Automatic Precise Positioning Service [90]. The CSRS PPP tool is particularly fast and easy to use.

2.1.0.1 Simple PPP Example

This section walks through a very simple, illustrative example of a PPP implementation using an extended Kalman filter [79]) and five dual-frequency satellites in view. Figure 2.1 shows a typical flow for an extended Kalman filter, which is a sequential filter, meaning that measurements are introduced as they are received over time rather than as a batch over multiple time steps. The filter maintains a state estimate and a covariance, which is the uncertainty on the state estimate. After initialization of the state and covariance, the time step does a simple kinematic propagation of the state and covariance to the current epoch. The measurement update uses the currently available pseudorange and carrier phase measurements to update the state and covariance.

The state vector used in this example is very simple and shown in Equation 2.1.

\[
\hat{X} = \begin{bmatrix}
x & y & z & b & \delta T & A^{(1)} & A^{(2)} & A^{(3)} & A^{(4)} & A^{(5)} \\
\end{bmatrix}^T
\]  

(2.1)

where:

- \(x, y, z\) : ECEF position
- \(b\) : clock bias
- \(\delta T\) : Tropospheric delay delta
- \(A^{(i)}\) : Float carrier phase ambiguity for satellite \(i\)

Notably, this state vector does not include the receiver velocity, but many common implementations do. The tropospheric delay delta term will be further expounded
2.1. PRECISE POINT POSITIONING

The measurements are used as dual-frequency iono-free combinations, so each satellite only has a single carrier phase ambiguity estimated, but uncombined implementations exist as well [29]. The position and clock bias elements of this state vector can be initialized with a simple least squares solution using code phase measurements. The tropospheric delay delta term can be initialized to zero with a sufficiently large covariance accommodate the variability. Similarly, it is often sufficient to initialize the carrier phase ambiguity estimates as zero, but depending on the receiver, it may be useful to initialize the carrier phase ambiguities with the difference between the code and carrier phase measurements from a particular satellite. Because the carrier phase ambiguity can be arbitrarily large, this ensures that the initial error is limited.

2.1.0.1.1 Time Update The time update propagates the state and the covariance from the previous epoch to the current epoch. The state and covariance are propagated by Equations 2.2 and 2.3, respectively. The subscripts to the state and the covariance indicate what epoch they represent and up to what time the information was used to create those values. For example, $\hat{X}_{t\mid t-1}$ can be read as the state at time $t$ given information from up to time $t - 1$. Equation 2.2 propagates the state
from time $t - 1$ to time $t$ with only measurement information from up to time $t - 1$.

State propagation:

$$\hat{X}_{t|t-1} = \Psi_t \hat{X}_{t-1|t-1} \quad (2.2)$$

Covariance propagation:

$$P_{t|t-1} = \Psi_t P_{t-1|t-1} \Psi_t^T + Q_t \quad (2.3)$$

The covariance, $P$ is propagated from time $t - 1$ to time $t$, and some process noise $Q$ is added to account for uncertainty in the dynamic model used. The dynamic model is represented by the state transition matrix (STM) $\Psi$.

$$\Psi = I \quad (2.4)$$

For this simple example, the STM, shown in equation 2.4, is simply the identity matrix. This represents a static model for the receiver position and clock bias, which may or may not be valid, so sufficient process noise must be added to the position and clock bias states to account for the potential change in position and clock bias over the time step interval. The process noise matrix, as shown in Equation 2.5, is a simple diagonal matrix. The values included in the matrix, in Table 2.1, indicate the level of correlation between the states at each time step. For the simple position kinematic model, a large amount of process noise is added to account for any potential velocity. This amount can be significantly reduced with a more representative dynamic model. Similarly, the clock bias process noise is large enough to accommodate large receiver clock rates. The tropospheric delta process noise is set for a nominal static model [39]. Finally, the carrier phase ambiguities are assumed to be static, so no process noise is added.
2.1. PRECISE POINT POSITIONING

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Process Noise Sigma [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_{pos}$</td>
<td>10000</td>
</tr>
<tr>
<td>$q_b$</td>
<td>10000</td>
</tr>
<tr>
<td>$q_{\Delta T}$</td>
<td>0.002/3600$^{1/2}$</td>
</tr>
<tr>
<td>$q_A$</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2.1: Example Process Noise

$$Q_t = \Delta t \begin{bmatrix}
q_{pos}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & q_{pos}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & q_{pos}^2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & q_b^2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & q_{\Delta T}^2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & q_A^2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & q_A^2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & q_A^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_A^2
\end{bmatrix} \tag{2.5}$$

2.1.0.1.2 Measurement Update  The measurement update is performed by Equations 2.6, 2.7, and 2.8.

Kalman Gain:

$$K_t = P_{t|t-1}H_t^T (H_tP_{t|t-1}H_t^T + R_t)^{-1} \tag{2.6}$$

State update:

$$\hat{X}_{t|t} = \hat{X}_{t|t-1} + K_t\tilde{y}_t \tag{2.7}$$

Covariance update:

$$P_{t|t} = (I - K_tH_t)P_{t|t-1} \tag{2.8}$$

Where $H_t$ is the sensitivity matrix, $R_t$ is the measurement noise matrix, and $\tilde{y}_t$ is a vector of measurement residuals. This can be broken down further, starting with the measurement residuals and the measurement model.
Measurement residuals:

\[ \tilde{y}_t = z_t - h(\hat{X}_{t|t-1}) \] (2.9)

The term \( z_t \) is the vector of observed dual frequency code and carrier phase measurements, and \( h(\hat{X}_{t|t-1}) \) is the non-linear measurement model of those measurements. In this example, the five pseudorange measurements comprise the first five elements of the measurement vector, and the five carrier phase measurements comprise the second half.

\[
\begin{bmatrix}
\rho(1) \\
\rho(2) \\
\rho(3) \\
\rho(4) \\
\rho(5) \\
\Phi(1) \\
\Phi(2) \\
\Phi(3) \\
\Phi(4) \\
\end{bmatrix}
\] (2.10)

The measurement model is where the specifics of PPP are incorporated. Careful modeling removes as much of the error as possible, and the carefully chosen estimated states mitigate the remaining errors.

Dual frequency carrier phase model:

\[
\Phi_{ij}^{(i)} = \| x_s^{(i)} - \hat{x}_{rx} \| + c(\hat{b}_{rx,c} - b_s^{(i)}) + T_{slant,\text{total}} + \Delta t_c + \Delta t_r + b_{\text{pau}}^{(i)} + \hat{A}^{(i)} + r_{st} + PCO_{rx} + PCV_{rx} + \epsilon^{(i)} \] (2.11)

Dual frequency code phase model:

\[
\rho_{ij}^{(i)} = \| x_s^{(i)} - \hat{x}_{rx} \| + c(\hat{b}_{rx,c} - b_s^{(i)}) + T_{slant,\text{total}} + \Delta t_c + \Delta t_r - DCB_{rx}^{(i)} - DCB_{sv}^{(i)} + r_{st} + PCO_{rx} + PCV_{rx} + \epsilon^{(i)} \] (2.12)
2.1. PRECISE POINT POSITIONING

where:

- \( x_s^{(i)} \): Precise satellite antenna phase center location
- \( b_s^{(i)} \): Precise satellite clock bias
- \( \Delta t_c \): Relativistic correction to clock bias
- \( \Delta t_r \): Relativistic correction to range
- \( r_{st} \): Earth solid tide adjustment projected onto receiver-satellite line of sight
- \( PCO_{rx} \): Receiver phase center offset adjustment
- \( PCV_{rx} \): Receiver phase center variation adjustment
- \( DCB_{rx}^{(i)} \): Receiver differential code bias
- \( DCB_{sv}^{(i)} \): Satellite differential code bias
- \( \epsilon^{(i)} \): Measurement noise

These terms are more fully described in Section 2.1.1. The sensitivity matrix, \( H_t \) is the partial derivative of the measurement model with respect to each of the estimated states. As such, the first three columns of \( H_t \) in Equation 2.13 are the “geometry matrix” as it is commonly referred to. The element \( G_x^{(i)} \) is the \( x \)-component of the line of sight vector to the satellite in ECEF [53]. The rest of the partial derivatives are fairly straightforward, as the clock bias and carrier phase ambiguities have a one-to-one mapping between their values and their impacts on the pseudorange and carrier phase measurements. The tropospheric mapping function is discussed in Section 2.1.1.4.

\[
H_t = \begin{bmatrix}
G_x^{(1)} & G_y^{(1)} & G_z^{(1)} & 1 & m_x^{(1)} & 0 & 0 & 0 & 0 \\
G_x^{(2)} & G_y^{(2)} & G_z^{(2)} & 1 & m_x^{(2)} & 0 & 0 & 0 & 0 \\
G_x^{(3)} & G_y^{(3)} & G_z^{(3)} & 1 & m_x^{(3)} & 0 & 0 & 0 & 0 \\
G_x^{(4)} & G_y^{(4)} & G_z^{(4)} & 1 & m_x^{(4)} & 0 & 0 & 0 & 0 \\
G_x^{(5)} & G_y^{(5)} & G_z^{(5)} & 1 & m_x^{(5)} & 0 & 0 & 0 & 0 \\
G_x^{(1)} & G_y^{(1)} & G_z^{(1)} & 1 & m_y^{(1)} & 1 & 0 & 0 & 0 \\
G_x^{(2)} & G_y^{(2)} & G_z^{(2)} & 1 & m_y^{(2)} & 0 & 1 & 0 & 0 \\
G_x^{(3)} & G_y^{(3)} & G_z^{(3)} & 1 & m_y^{(3)} & 0 & 0 & 1 & 0 \\
G_x^{(4)} & G_y^{(4)} & G_z^{(4)} & 1 & m_y^{(4)} & 0 & 0 & 0 & 1 \\
G_x^{(5)} & G_y^{(5)} & G_z^{(5)} & 1 & m_y^{(5)} & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\]
The measurement noise matrix, $R_t$, like the process noise matrix, is a simple diagonal matrix. The dual-frequency code phase noise, $\sigma_\rho$, used in this thesis is 2 meters, and the dual-frequency carrier phase noise is set to 0.03 m. These are both inflated relative to what they would be if used as single-frequency measurements [53], and can vary based on the receiver and antenna pair used to produce the observations.

$$R_t = \begin{bmatrix} \sigma_\rho^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_\rho^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_\rho^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_\rho^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_\rho^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_\Phi^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_\Phi^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_\Phi^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_\Phi^2 \end{bmatrix}$$

(2.14)

Once each of these elements is created, the measurement update can finally occur, bringing the filter fully up to the current time.

Figure 2.2 shows the result of the PPP algorithm for a static antenna on the roof of the Stanford Aeronautics and Astronautics building. The $2 - \sigma$ value of the covariance in the horizontal direction is plotted alongside the error in the position state. The covariance drops significantly initially and continues to converge as the sensitivity matrix changes due to the relative movement of the satellites overhead. After approximately 30 minutes, the solution has converged, and the error is on the order of centimeters.

### 2.1.1 Modeling of GNSS Observations

This section offers further description of the models used in PPP pseudorange and carrier phase prediction.
2.1. PRECISE POINT POSITIONING

2.1.1.1 Precise Satellite Orbit and Clock

One of the required products for PPP is precise orbit and clock information. The precise orbit and clock data used in this thesis are provided by the analysis centers of the International GNSS Service’s (IGS) Multi-GNSS Experiment (MGEX) project. MGEX data is not available in real-time. There is a latency of typically greater than one week, but for post-processing, these products are very useful. MGEX claims GNSS signal-in-space ranging error (SISRE) of 5 cm or better root mean squared for GPS, GLONASS, and Galileo, which is accurate enough to enable PPP [58]. The MGEX analysis centers use a global network of hundreds of survey-grade multi-GNSS receivers in conjunction with high fidelity force models to produce extremely accuracy orbital estimates.

Figure 2.2: Static PPP Solution Example
The data is hosted on various publicly available websites and is in the RINEX .sp3 standard format [38]. Orbit data is provided at the center of mass of the satellite in ECEF at 5- or 15-minute rates. In the PPP algorithm, satellite positions need to be available at the time of transmission of the signals, so some interpolation of the precise data must be done. Several techniques are available for such interpolation, including polynomial, Lagrange, and trigonometric interpolation [73] [32]. An 11th order Lagrange interpolation is used for its millimeter accuracy and ease of implementation. Center of mass data provided at 15-minute intervals is interpolated to the desired epochs, at which point the offset between the center of mass and the antenna phase center is applied, which will be discussed in the next section.

MGEX satellite clock products are typically produced at much higher rates than their orbital counterparts due to the much more unpredictable nature of on-orbit atomic clocks at the sub-nanosecond scale. Simple linear interpolation is typically a sufficient method for precise clock usage, as the dynamics of clocks are quite simple, and only a clock bias, drift, and sometimes drift rate are considered. An important note on precise clock products is that they refer to a specific ionosphere-free dual frequency signal combination. Just as the GPS navigation message clock state is referenced to the L1 P(y) – L2 P(y) combination and a timing group delay (TGD) term must be applied for L1 C/A usage, similar differential code biases must be used for single frequency measurements or dual frequency measurements that are not the same as the reference signal pair.

2.1.1.2 Antenna phase center offset and variation

IGS satellite orbital estimates refer to the center of mass (CoM) of the satellite, but ranging signals are broadcast from the phase center of the transmission antenna. In order to reconcile these two positions, a body-frame displacement between the CoM and the antenna phase center (APC) and a body frame attitude model are required. The IGS publishes antenna phase center offsets in the body frame for all GNSS satellites. Such offsets can be greater than one meter in the boresight direction, making this a very important correction. The attitude of the satellite is such that the solar panels are always pointed to maximize received sunlight [54]. The nominal
attitude model is described by a right handed system as follows:

\[ \hat{e}_x = \hat{e}_y \times \hat{e}_z \quad (2.15) \]

\[ \hat{e}_y = \frac{(\hat{e}_{\text{sun}} \times \vec{r}_{sv})}{(\hat{e}_{\text{sun}} \times \vec{r}_{sv})} \quad (2.16) \]

\[ \hat{e}_z = \frac{\vec{r}_{sv}}{\vec{r}_{sv}} \quad (2.17) \]

Where \( \hat{e}_{\text{sun}} \) is an ECEF unit vector pointing to the sun, and \( \vec{r}_{sv} \) is an ECEF position vector of the vehicle. \( \hat{e}_x, \hat{e}_y, \) and \( \hat{e}_z \) describe a right-handed body frame of the satellite.

The position of the sun can be computed to the necessary accuracy here using orbital elements representing the solar ephemeris. Even over long periods and including the rotation of the earth, a simple ephemeris can provide a solar unit vector that matches those produced by higher fidelity products to the milliradian level.

It is important to note that during periods of eclipse, most GNSS satellites’ attitudes cannot be reliably determined by the nominal attitude model, and different blocks of satellites have different eclipse behavior. The PPP filter built for this thesis does not consider eclipse behavior and instead discards the data associated with satellites in eclipse.

Antenna phase center variation (PCV) is a look angle and frequency dependent bias that can be observed in GNSS carrier phase measurements. The IGS also provides products that provide the PCV as a function of elevation and azimuth from the satellite antenna frame that can be interpolated. Such variations are typically sub-cm.

2.1.1.3 Receiver phase center offset and variation

Just as the satellite antenna phase center is not located at the center of mass of the satellite, the receiver antenna phase center may not be collocated with the body frame reference position of the estimator. In this case, a lever arm offset must be applied, which is discussed Section 2.2. There may also be a frequency dependence on the
antenna phase center offset as compared to a physical reference point on the antenna. For a survey-grade antenna, this position difference is typically millimeter level. Similarly, the phase center variation of survey-grade antennas are in the millimeter range. For antennas in the IGS network, estimates of the antenna phase center offset and variation are provided in a similar fashion to those of the GNSS satellites. For other receiver-antenna setups, the choice of antenna becomes very important. For precision applications, using an inexpensive patch antenna may not suffice because they can have frequency dependent offsets and variations on the order of 10-20 centimeters [28]. These can impact the solution significantly, especially if ambiguity fixing is desired.

2.1.1.4 Tropospheric delay

The Earth’s troposphere introduces a non-dispersive, i.e. unimpacted by signal frequency, delay on GNSS signals. The total tropospheric delay can be separated into hydrostatic, or dry air, and wet components, and a common method of modeling tropospheric delay is to compute a zenith tropospheric delay that is then mapped to the slant delay at the given elevation and azimuth. In the case of PPP, an additional estimated term is introduced, so the total tropospheric delay can be modeled as follows:

\[ T_{\text{slant, total}} = m_d(E)D_{Z,D} + m_w(E)D_{z,w} + m_w(E)\delta T \]  \hspace{1cm} (2.18)

where:

- \( E \) : Elevation angle of satellite
- \( m_d \) : Hydrostatic mapping function
- \( D_{Z,D} \) : Hydrostatic modeled zenith delay
- \( m_w \) : Wet mapping function
- \( D_{z,w} \) : Wet modeled zenith delay
- \( \delta T \) : Estimated delta tropospheric zenith delay

The PPP implementation in this thesis uses the UNB3 tropospheric model [52]. The UNB3 model is empirically driven, where the zenith wet and dry delays as well as the mapping function are functions of user elevation, latitude and day of year. The modeled dry delay is much larger than the modeled wet delay, but the dry delay varies much less than the wet delay over the course of a year for a given location.
2.1. PRECISE POINT POSITIONING

Figure 2.3: UNB3 zenith dry delay time variation

Figure 2.3 shows the zenith dry delay as function of day of year and latitude. At sea level, the dry delay is approximately 2.3 meters and varies very little. The wet delay, while much smaller on average, varies much more based on seasonal conditions, as shown in Figure 2.4.

As user altitude increases and the amount of troposphere that the signal must pass through decreases, both the zenith dry and wet delays decrease significantly. In fact, the UNB3 modeled wet delay decreases to nearly 0 by 5000 meters of user altitude, and the corresponding dry delay has nearly halved. For airborne GNSS users, it is important to increase the process noise added to the estimated tropospheric delay term, as the nominal model assumes that the local atmospheric conditions only change slowly, whereas a fast moving user may actually move into new weather conditions.
Figure 2.4: UNB3 zenith wet delay time variation

The mapping function used is the mapping of Niell [61], which is normalized such that at zenith, the zenith tropospheric delays are unaffected. Figure 2.6 shows that at low elevations, they can be increased by a factor of five or more.

2.1.1.5 Satellite Differential Code Biases

Differing processing and RF chains on the satellite introduce varying delays per GNSS signal [55]. These varying delays can be described as satellite differential code biases (DCB). Delays are variable across not only broadcast frequencies but also across signals even on the same frequencies. For example, L1 C/A and L1 P code phase measurements can be found to have different line delays, and that delay is different per PRN. For a float PPP user, it is important to calibrate these delays, which can
be at the meter-level, before using code phase measurements in the filter. Otherwise, the position solution will be biased by the ranging biases.

IGS ACs produce estimates of DCBs in different forms. The German Aerospace Center (DLR) produces multi-GNSS DCBs that are valid for a week and published in three-month intervals. While these may be less appropriate for real-time or near-real-time applications, they are perfectly acceptable for the post-processed work in this thesis.

2.1.1.6 Receiver Differential Code Biases

Just as differing processing and RF chains on GNSS satellites can introduce varying code delays, those effects are present at the receiver as well. Typically, when using
only a single frequency or dual-frequency pair, the code delays are absorbed into the receiver clock bias estimate. When more than two frequencies are used, if receiver DCBs are not taken into account, the various pseudoranges will not be consistent. To mitigate this, receiver DCBs can be estimated. One code phase signal, whether single frequency or iono-free, can be taken as a reference and DCBs estimated for the other signals [39]. For GPS and Galileo, a single DCB can be estimated across all PRNs for a particular signal, but because GLONASS uses a frequency division multiple access (FDMA) scheme, many PRNs are on different frequencies. Even these small frequency offsets on what is nominally L1 can introduce variable code phase delays. The mitigation strategy for GLONASS then becomes to take a particular FDMA channel as the reference signal and to estimate DCBs for each of the other
channels. In all cases, the receiver DCBs are assumed to be static, and no process noise is added.

2.1.1.7 Ionospheric Delay

The charged upper atmosphere, the ionosphere, is a dispersive medium that introduced variable delays or advances on the GNSS code and carrier measurements that are a function of the frequency of the signal. Measurements from multiple frequencies can be combined such that up to 99.9% of the ionospheric delay (or advance, as is the case on the carrier phase measurements) can be removed \[65\]. This is primarily the method of ionospheric mitigation taken in this thesis. An alternative is to use externally estimates of the ionospheric delays in combination with single frequency measurements.

2.1.1.8 Relativistic Effects

For float PPP, two relativistic effects should be modeled. The first relates to the variation in the satellite clock rate as it travels in a non-circular orbit. The resulting clock bias offset, which can be meter-level, can be described by Equation 2.19.

\[
\Delta t_c = \frac{2 \vec{r}_{sv} \cdot \vec{v}_{sv}}{c^2}
\]

(2.19)

where:
\(\vec{r}_{sv}\) : ECEF satellite position
\(\vec{v}_{sv}\) : ECEF satellite velocity
\(c\) : speed of light

The Shapiro time delay introduces increased path length of up to 1-2 cm based on the elevation angle of the satellite. This effect is caused by the signal traveling through the gravitational potential of the Earth and can be described by Equation 2.20.

\[
\Delta t_r = 2 \mu_{earth} \frac{\log |\vec{r}_{sv}| + |\vec{r}_{rx}| + |\vec{r}_{sv} - \vec{r}_{rx}|}{|\vec{r}_{sv}| + |\vec{r}_{rx}| + |\vec{r}_{sv} + \vec{r}_{rx}|}
\]

(2.20)

Where \(\mu_{earth}\) is the Earth’s gravitational constant.
Figure 2.7: Radial displacement of Earth Solid Tides over a day
2.1.1.9 Phase Windup

The carrier phase measurements in particular are affected by the change in attitude of the satellite. As the transmitting satellite yaws to maintain exposure of the solar panels to the sun, the right hand circularly polarized signal that is received experiences an apparent change in phase. This effect is described in detail in [86].

2.1.1.10 Earth Solid Tides

The gravitational pull of the sun and the moon deform the Earth’s crust on the order of tens of centimeters on a daily basis [63]. Figure 2.7 shows the displacement from a nominal geoid in the radial direction over a single day. The variation is a function of the time of day, latitude, longitude, and the position of the sun and the moon in ECEF. The effects of solid tides are highly predictable but must be taken into account for cm-level positioning.

2.2 Tightly Coupled Inertial Measurement Unit

PPP is a very powerful technique, but as is the case with any technique reliant on GNSS, it can be subject to measurement availability and continuity issues in environments that are not fully open sky. In order to mitigate these effects, one can include other sensors into the integrated navigation solution. Inertial measurement units, comprised of a 6 degree of freedom accelerometer and gyroscope, complement the GNSS solution very well. IMUs are themselves subject to a variety of errors, but the GNSS measurements can be used to estimate them. At the same time, the IMU is not impacted by overhead blockages that affect the GNSS measurements and can be used to bridge gaps in GNSS availability. This section serves as only a high-level overview of the integration of a PPP filter with an IMU in a tightly coupled fashion. For much more detail, [33] and [43] offer great insight and specifics.
The tightly coupled IMU-PPP structure remains largely the same as the case without the IMU, except for the addition and modification of a few states and the inclusion of the IMU mechanization for state propagation. The primary role of the IMU is to propagate the position and velocity of the navigation system using the measurements from the accelerometer and the gyroscope. The propagation of the IMU position, velocity, and attitude given accelerometer measurements, gyroscope measurements, and a gravitational model is commonly called the mechanization. One of the new states required is the body frame attitude of the IMU, which is necessary because the IMU measurements are taken in the IMU body frame, which must then be mapped to the navigation frame, which is in this case ECEF. The attitude is maintained as a direction cosine matrix (DCM) as well as an error state vector of body frame Euler angles \[8\]. The use of Euler angles is not subject to gimbal lock because they are used as error states and as such are always small angles. Even high
quality IMU’s are subject to errors in the form of biases or scaling issues. As such, the tightly coupled structure can be used to estimate these errors to greatly improve the performance of the propagation. A very common set of errors to estimate are simply slowly varying biases on the accelerometer and gyroscope outputs. A bias is estimated in each direction for the gyroscope and the accelerometer. The position and velocity states must also be modified. In a PPP-only system, the estimated position and velocity are often that of the GNSS antenna phase center. When an IMU is introduced, it is common to instead estimate the position and velocity at the IMU for ease in the mechanization. Because of this, the state at the IMU must be able to be translated to the state at the APC given a surveyed lever arm offset in the body frame as well as the attitude and attitude rate of the IMU. The relative position of the IMU and APC must be well known for precise propagation.

Figure 2.8 shows the tightly coupled IMU-PPP block diagram. It is largely the same as without the IMU, except for the IMU and mechanization blocks. The primary change comes in the propagation of the position, velocity, and attitude states in the mechanization block. It is typical for an IMU to output measurements at a higher rate than the GNSS receiver. For example, an IMU may produce measurements at 100 Hz, while the GNSS receiver only operates at 1 Hz. The IMU mechanization block propagates the position, velocity, attitude, and IMU bias states using the high-rate IMU measurements of specific force and angular rate. This occurs repeatedly, represented by the inner loop going from the mechanization block back into itself, until a GNSS measurement is available. At this point, the time update can begin. While in a PPP-only case, the position and velocity state would be propagated to the current time step in the time update, the IMU mechanization has already performed the propagation. Instead, only the GNSS-specific states, such as the clock bias and tropospheric delta states must be propagated. The construction of the state transition matrix must still occur for all of the states and is very important. The state transition matrix captures the correlations between the various state errors as they are propagated forward in the mechanization. It is through the STM and thus the covariance that the various IMU states are connected in the filter.

In the measurement update, the GNSS measurements are handled. As previously
mentioned, the position and velocity used as inputs to the measurement models must be modified to account for the lever arm offset. While the measurements are not very (or in some cases not at all) sensitive to the attitude, accelerometer, and gyroscope bias states, the measurement update still results in updated estimates of essentially all of the states. Again, this is because the STM in the time update builds the connection between the states. The result of the measurement update is an improved state estimate that is then fed back into the mechanization until the next time step.

2.3 Advanced Receiver Autonomous Integrity Monitoring (ARAIM)

GNSS measurements serve as the basis for many navigation methods, but they can be subject to faults. Faults are large, unexpected errors in the measurements. Fortunately, the multitude of satellites in view at any given time for most GNSS users allows for redundant measurements that can be used for consistency checking and error bounding. One such technique to do so is called receiver autonomous integrity monitoring (RAIM). RAIM has been used for en route navigation for decades and is able to provide horizontal protection levels of one nautical mile with high availability using only GPS L1 signals [6] [17].

With new GNSS constellations coming online, there is an opportunity to leverage the additional satellites and signals for high integrity navigation [40]. Additional satellites improve the geometry of the navigation solution while also increasing the number of redundant measurements for fault detection. Full constellations of dual frequency capable satellites allow for the cancellation of the ionospheric error, which significantly reduces the nominal ranging errors. New constellations also introduce risks of new fault types and higher fault rates. Advanced RAIM seeks to exploit these opportunities while mitigating risks to extend the capabilities of RAIM to allow for vertical guidance [20] [62]. Localizer performance with vertical guidance (LPV) are high precision GPS aviation instrument approach procedures that are currently enabled by satellite based augmentation systems (SBAS). Each SBAS is a regional
service that provides low-latency corrections and integrity parameters for GPS. The most stringent operation targeted by ARAIM is LPV-200, which allows for guidance all the way down to 200 feet above the height of the runway. This requires a Vertical Alert Limit (VAL) of 35 meters, where the alert limit is the maximum allowed protection level for the service. The probability that the position error exceeds the alert limit must be less than $10^{-7}$ per approach, and any errors must be flagged for the pilot within six seconds of onset. ARAIM will provide service for even these cases with strict requirements. ARAIM requires less infrastructure than SBAS and is a global, rather than regional, service. A joint US-EU effort has been created to develop ARAIM through the Working Group C (WG-C) ARAIM Technical Subgroup (ARAIM TSG) [1], [2].

The ARAIM architecture requires that statistical parameters relating to the nominal errors and probabilities of fault of the GNSS measurements be disseminated to the aviation users periodically. These parameters will be disseminated in the form of an integrity support message (ISM) and will allow for the user algorithm to autonomously produce protection levels and perform fault detection and exclusion. The primary integrity algorithm discussed in this section is multiple hypothesis solution separation (MHSS), which compares navigation solutions produced using different subsets of the available measurements [5]. This section offers a brief description of the ARAIM concept and architecture.

### 2.3.1 Multiple Hypothesis Solution Separation

The real-time ARAIM user algorithm, given GNSS code and carrier measurements with their corresponding navigation messages, computes a position solution, computes protection levels, and performs fault detection and exclusion at each epoch [62], [17], [11], [13] [23]. The estimator used in ARAIM is a snapshot least squares estimator. Protection levels provide a statistical bound on the error on the position solution.

$$PHMI = P(|x - \hat{x}| > PL, |q| < T)$$  \hspace{1cm} (2.21)
where: \( PHMI \): Probability of hazardous misleading information or the integrity risk allocation
\( x \): the true state
\( \hat{x} \): the estimated state
\( PL \): the protection level
\( |q| \): the test statistic
\( T \): a threshold based on false alarm requirements and subset covariances

Essentially, the integrity risk is the probability that the position error exceeds the protection level and that the test statistics are below the threshold. The ARAIM approach uses multiple hypothesis solution separation (MHSS) to break down the integrity risk into components based on the fault hypotheses, so Equation 2.21 can be rewritten:

\[
PHMI = \sum_{i=0}^{H} P(|x - \hat{x}| > PL, |q| < T|H_i)P_{H_i} \tag{2.22}
\]

Each fault hypothesis \( H_i \) contributes to the integrity risk individually, where each fault hypothesis corresponds to a fault or set of faults given the available measurements. \( P_{H_i} \) is the prior probability of fault mode \( i \), which can be evaluated based on the \( P_{sat} \) and \( P_{const} \) disseminated in the ISM, which is discussed in a following section. If any of the test statistics exceed their corresponding thresholds, a detection is declared.

The test statistic for each fault hypothesis is the position difference between the subset solution and the all-in-view (AIV) solution. The AIV solution uses all available GNSS measurements. Each subset solution is generated by excluding the measurements corresponding to the fault hypothesis it monitors. By excluding measurements, each subset is robust to and will be unaffected by faults occurring on the excluded measurements. For example, if eight fault modes are monitored, nine total position solutions are generated including the AIV solution. The test statistics are then the
position differences between each of the subset solutions and the all in view. Furthermore, the protection levels can be computed for any particular desired spatial direction, in which case, the test statistic is the subset position difference in that direction.

The continuity requirements and subset covariances are used to compute the detection thresholds, where the detection threshold is set to limit the probability of false alarm for that fault mode. The smaller the allocated probability of false alarm for a given fault mode, the larger the detection threshold must be. The threshold is also a function of the subset covariance, which is produced by the position estimator.

ARAIM requires that the AIV and subset covariances be properly calibrated such that they bound the nominal, unfaulted errors. This is done through the careful construction and characterization of the nominal pseudorange error model:

\[ \sigma_{\text{integrity},i}^2 = \sigma_{\text{URA},i}^2 + \sigma_{\text{user},i}^2 + \sigma_{\text{tropo},i}^2 \]  

(2.23)

The \( \sigma_{\text{URA}} \) corresponds to the broadcast user range accuracy (URA) value scaled by the ISM. \( \sigma_{\text{tropo},i} \) is computed from a physical model, and ionospheric error is neglected for the dual frequency user. \( \sigma_{\text{user},i} \) captures additional local effects including the smoothed multipath error and an additional noise term.

### 2.3.2 System Architecture

Two architectures have been considered to determine values of the ISM and disseminate them to the user receivers: “online” and “offline.” The primary difference between the two architectures lies in the frequency of ISM updates as well as an additional ephemeris overlay provided in the online approach [19]. In the online approach, a dedicated ground network of monitoring stations allows for the real-time generation of the ISM as well as an ephemeris overlay that augments the broadcast navigation message. The online approach requires a significant amount of infrastructure to not only produce the ISM but to disseminate it in near real-time and is similar to SBAS or GBAS in its overall approach. The offline approach, which is what is considered in this thesis, has a much longer latency to the ISM updates and no ephemeris overlay.
Instead, the ISM parameters are held generally static, which requires that the constellation performance is similarly static. This necessitates trust from the air navigation service provider’s (ANSP) that the constellation performance will remain as good or better than it has been in the past. Moving forward, this thesis will only consider the offline architecture.

Figure 2.9: ARAIM Offline Monitoring Architecture [1]

Figure 2.9 shows the offline architecture. The ISM generation is done by collecting GNSS observations through a ground network of receivers over a long period of time. Those observations are used to compare GNSS performance against the performance commitments provided by the constellation service provider (CSP). The ISM generated from the long-term statistical analysis done with the collected monitoring data is then disseminated to the airborne receivers for use in the ARAIM algorithm. The CSP performance commitments, published in a performance standard, must clearly describe several different aspects, including the nominal ranging accuracy, the probability of fault, a description of such possible faults, the expected time to alert users of faulted performance, and the expected service availability.
2.3. ADVANCED RECEIVER AUTONOMOUS INTEGRITY MONITORING (ARAIM)

2.3.3 Integrity Support Message

The ISM contains parameters that should conservatively bound the ranging errors observed by the airborne receiver such that even if no new ISM were to be disseminated to the receiver, the latest values could safely bound errors in the future. However, the values should not be overly conservative such that performance is diminished unnecessarily. The ISM parameters pertain to nominal ranging errors as well as the probability of larger errors, or faults. The threats that the ISM covers are related to the signal in space, including: broadcast clock and ephemeris errors, signal deformation errors, code and carrier incoherence, and differential code and satellite antenna biases [10]. The primary threats that are examined in this thesis are related to the broadcast clock and ephemeris in Appendix A and differential code biases in Chapter 5.

Table 2.2 shows some of the key ISM parameters. The first three parameters, $\alpha_{URA,j}$, $\alpha_{URE,j}$, and $b_{nom,j}$, relate to the nominal ranging performance from satellite $j$. $\alpha_{URA,j}$ and $\alpha_{URE,j}$ are multipliers to the broadcast $\sigma_{URA}$ that allow for the airborne receiver to inflate the URA terms broadcast for integrity and continuity, respectively. The $\alpha$ parameters can be determined through the careful examination of long term ranging errors. In particular, the characterization of the always-changing broadcast clock and ephemeris errors is important to the determination of the $\alpha$ parameters [84]. $b_{nom,j}$ describes the nominal expected ranging bias, whether caused by signal deformations or a static offset in the broadcast clock and ephemeris parameters over long periods. Finally, the $P_{const,i}$ and $P_{sat,j}$ are the probabilities that the SIS ranging error is larger than a threshold set by the CSP. $P_{const,i}$ is the probability that multiple satellites experience a fault simultaneously due to the same underlying cause. This is known as a constellation or wide fault. The cause of a wide fault could be from the upload of bad clock or ephemeris parameters to multiple satellites or other system-wide failures. Similarly, $P_{sat,j}$ is the probability that a single satellite experiences large errors independently from the other satellites and is known as a narrow fault. Narrow faults have been observed to be caused by on-orbit clock or other hardware malfunctions as well as the upload of erroneous navigation data [83].

The evaluation of $\alpha_{URA,j}$, $\alpha_{URE,j}$, $P_{const,i}$, and $P_{sat,j}$ can be done through the long
 CHAPTER 2. BACKGROUND

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_{URA,j}$</td>
<td>URA multiplier for integrity</td>
</tr>
<tr>
<td>$\alpha_{URE,j}$</td>
<td>URA multiplier for continuity and accuracy</td>
</tr>
<tr>
<td>$b_{nom,j}$</td>
<td>Nominal bias term [m]</td>
</tr>
<tr>
<td>$P_{const,i}$</td>
<td>Probability of constellation (wide) fault at a given time</td>
</tr>
<tr>
<td>$P_{sat,j}$</td>
<td>Probability of satellite (narrow) fault at a given time</td>
</tr>
</tbody>
</table>

Table 2.2: Key ISM Parameters

term comparison of the broadcast ephemeris and clock navigation messages to post-processed precise ephemeris and clock estimates. The offline ARAIM architecture does not necessarily need a dedicated ground monitoring infrastructure, and the IGS data repositories offer much of the desired data. The IGS provides raw observations of the pseudorange and carrier phase as well as navigation data logs from its network of hundreds of multi-constellation receivers. Perhaps most importantly, the IGS analysis centers (ACs) produce centimeter accurate estimates of the satellite orbit and clock states. Figure 2.10 shows the general structure of the evaluation of the broadcast orbit and clock. The difference is taken between the post-processed states and the state output from the GNSS navigation message curve fit for each satellite over long periods of time. Those state errors are projected onto a representative grid of users in the terrestrial volume. That line of sight error can be normalized by the broadcast $\sigma_{URA}$ and the resulting distribution overbounded to determine values for $\alpha_{URA,j}$ and $\alpha_{URE,j}$. Those normalize line of sight error values can also be used to identify narrow and wide faults. This process is described in more detail in Appendix A.

Figure 2.10: Constellation monitoring structure
Chapter 3

PPP with Integrity

3.1 Introduction

UAV and autonomous platforms can greatly benefit from an assured position solution with high integrity error bounds. The expected high degree of connectivity in these vehicles will allow users to receive real time precise clock and ephemeris corrections, which enable the use of precise point positioning techniques. PPP techniques [50] can provide centimeter accuracy without local reference stations in kinematic applications. These techniques have so far mostly been used to provide high accuracy, and it is only recently that they have been proposed to provide integrity, that is, position error bounds with a very low probability of exceeding them. There has been preliminary work [60] on the application of integrity to PPP, but it remains a challenge to translate the benefits of PPP to accuracy while maintaining high integrity. Most of the integrity work in PPP and real-time kinematic (RTK) has dealt with the ambiguity resolution process under nominal error conditions [74], [26], and less on the integrity of the position solution under fault conditions.

This chapter shows that PPP, in conjunction with techniques developed for integrity in aviation, can be used to produce meter-level protection levels for static, automotive, and flight scenarios, where the probability of the true error exceeding the protection level is $10^{-7}$. Solution separation, which requires the creation of solutions that are tolerant to faults or sets of faults, is shown to provide integrity to the
CHAPTER 3. PPP WITH INTEGRITY

PPP algorithm under specific assumptions (i.e. bounded probabilities of measurement faults). This chapter also builds off of this to improve a PPP with solution separation algorithm by using an inertial measurement unit (IMU) to improve solution and protection level continuity.

3.2 Prior Art

The prior art below seeks to provide a safe navigation solution by either producing a protection level or by alternatively either rejecting faulted measurements or inflating measurement covariances to account for additional error.

- GMV has created the patented isotropy based protection level (IBPL) [72] as a method to bound errors for single epoch least squares solutions. This method makes the assumption that the measurement error vector is equally likely in every direction in the measurement space, which is not the case for most other methods. The also-patented kalman integrity protection level (KIPL) [7] is an extension to the IBPL, where the residuals from each measurement are used to estimate error distributions that are then combined to map to an estimate error distribution that can be bounded. This method has been demonstrated in many environments, but as it is patented, it is not able to be used by others without payment, and some details as to the implementation are excluded from the patent. Additionally, the KIPL does not offer a fault detection and exclusion function.

- Solution separation has been applied to a kalman filter GNSS solution [21], but precise estimation algorithms are not used and protection levels of less than 10 meters are not achieved.

- Advanced RAIM [17] builds a basis for the integrity methods in this thesis, but it is built for snapshot estimation with broadcast navigation messages and also does not offer protection levels of the desired magnitude.

- A number of efforts have also built from the aviation community towards using
sequential filters [45] or batch approaches [46]. These efforts are also not driving for high accuracy, and they stop at the level of covariance analyses, so they are insufficient for the level of desired performance.

- In urban environments, measurements may not behave according to the nominal pseudorange models developed for RAIM and ARAIM. Gaussian mixture models can be used to model pseudoranges as the weighted sum of multiple distributions [59]. This is particularly powerful in the case of non-line of sight multipath, but this can encounter issues when the state of the signal changes, i.e. from line-of-sight (LOS) to non-line-of-sight (NLOS).

- A random sample consensus (RANSAC) based approach is taken in [24], where rather than subsets excluding particular satellites, subsets instead add new satellites such that the solutions are consistent. However, this method is even more computationally expensive and does not come with a robust proof of safety.

- This thesis deals largely with the integration of multiple sensor types in a high integrity solution. Other works have integrated more unique techniques such as the use of map matching [81] or low cost IMU and a fish eye lens [75]. In the case of map matching, especially with complex 3D urban environment models, computational cost can be overly burdening. The fisheye lens offers an additional method for NLOS signal classification, but it does require an upward facing camera.

- Many more integrity techniques related to GNSS-based navigation in urban environments can be found in [89].

### 3.3 Filter Description

This section steps through Figure 3.1 to describe the implementation of the PPP/IMU extended kalman filter with solution separation. Beginning with the ingestion of new measurements, subsets and states are added and/or removed as necessary. Next, the time update propagates the banks of states and covariances to the current epoch.
The measurement update is performed with special care to reduce the computational complexity and remove immediately obvious outliers. This will be explained in more detail later. Finally, protection levels are computed and faults potentially detected based on the states and covariances produced by the bank of filters.

![PPP with Solution Separation Implementation Diagram](image)

**Figure 3.1: PPP with Solution Separation Implementation Diagram**

### 3.3.1 Subset Management

The number of subsets in use is one of the primary drivers of computational load, so care must be taken to use the minimum number of subsets needed. The simplest subsets to monitor are so called one-out subsets, where each subset excludes a single satellite, each with its own probability of fault. If the probability of fault for individual fault modes is high enough, then it’s possible that fault modes consisting of multiple simultaneous faults must be considered, e.g. a fault mode consisting of satellite 1 and satellite 5 faults at the same time. The number of subsets monitored is a combinatorial problem, so the number of fault modes is very important if simultaneous faults can occur.

To reduce the computational load, single fault modes can be grouped together to
3.3. FILTER DESCRIPTION

Figure 3.2: Grouping fault modes in a single subset reduces total subset count

monitor faults on either satellite. In this case, each subset monitors faults from multiple satellites, as shown in Figure 3.2. This must also be reflected in the probability of satellite fault for the new fault mode that covers measurements from either satellite. Grouping faults can significantly reduce the number of subsets in the “two-out” case. If only single satellite faults are considered, grouping subsets into pairs halves the number of monitored subsets. If two-out subsets are considered, the effect is much more profound, as is shown in Figure 3.3. The accompanying negative effect is weakened geometries in the subsets and in the two-out subsets in particular, which will typically lead to larger protection levels. This can be mitigated to an extent through the use of an IMU, as shown in [35].

The procedure for subset handling is as follows and is performed at the beginning of every time step:

1. Given current set of available GNSS observations, add newly available satellites to existing list of subset groups.

2. Form full list of required subsets given the probability of fault of each of the subset groups and the target integrity risk.

3. Spawn new subsets given the newly formed list of required subsets and the existing subsets. New subsets are initialized using the all-in-view solution from the previous time step.

Following this procedure over very long periods would result in subsets of all
possible satellites, which is undesirable because most of the satellites are only in view for a period of a few hours, and it seems reasonable to not monitor faults from satellites that are not currently being observed. However, the filter solution is still dependent on the previous measurements from the set satellite, so the subset excluding the set satellite is still required. In order to keep from accumulating subsets indefinitely without removing any, the proposed strategy is to simply reinitialize the AIV and subset filters periodically and run these filters in the background until they have sufficiently converged. When the reinitialized filters have converged, the primary set of filters can be removed, and the reinitialized filters can be promoted to become the primary set of filters. The timing of the choice to switch to the reinitialized filters can be purely driven by performance; if the reinitialized filters are converged to the desired protection levels or if the user no longer wishes to carry double the number of subsets, the reinitialized filters can be swapped in for the old primary filters. The spawning of reinitialization subsets can be triggered in several ways. The simplest trigger is on the swap of reinitialized filters to primary filters. In this mode, there is always a
second set of filters in the background, and they can be promoted periodically. This limits the maximum exposure time for a given fault to twice the convergence time of the filters, which for multi-constellation PPP with good visibility can be on the order of a few minutes. Another trigger could be, as hinted at previously, the setting of a satellite.

### 3.3.2 State Management

Because the states are specific to individual satellites and signals, they need to be managed as satellites come into or out of view. There are a few cases that need to be considered based on the currently available measurements: when a new satellite or signal appears to the filter, when a previously available satellite is no longer available, and when a cycle slip is detected.

When a new set of GNSS measurements is available, the filter adds states as necessary for measurements from satellites or signals that were not previously tracked in the filter. For example, if PRN 5 rises above the elevation threshold, carrier phase and code phase multipath states are added to the state vectors across all subsets. The initial values are set to zero, and the initial uncertainties are set according to the values in Table 3.1. Similarly, when signals or satellites are no longer visible that were previously available, those states are immediately removed from the state vector. Cycle slips are identified either by a simple geometry free check when multiple frequencies are available or the receiver cycle slip detection indicator.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Initial Sigma</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tropospheric Wet Delay</td>
<td>0.1</td>
<td>m</td>
</tr>
<tr>
<td>Float carrier phase ambiguity</td>
<td>100</td>
<td>m</td>
</tr>
<tr>
<td>Carrier phase error state</td>
<td>0.05</td>
<td>m</td>
</tr>
<tr>
<td>Code Phase Multipath</td>
<td>2</td>
<td>m</td>
</tr>
<tr>
<td>Differential Code Bias</td>
<td>20</td>
<td>m</td>
</tr>
<tr>
<td>Frequency-Dependent DCB</td>
<td>1</td>
<td>m</td>
</tr>
</tbody>
</table>

Table 3.1: Initial State Uncertainties
3.3.3 Time Update

The time update serves to propagate the error state and covariance to the current time step. The filter can run in two different modes— with and without an IMU. Whether or not the IMU is used makes a significant impact on the time update. When the IMU is not used, the position and velocity propagation is done according to a very simple kinematic motion model with the following state transition matrix (STM) for a state vector consisting of just a position and velocity:

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

(3.1)

This STM is used to propagate the reference trajectory as well as the covariance. The process noise added to the position and velocity states are described in Table 3.2. The process noise added to the position and velocity propagation is chosen such that a smaller amount is added in the vertical direction so as to constrain the error growth for reasonable automobile dynamics. When an IMU is incorporated, the position, velocity, attitude, and bias propagation become more involved. The position, velocity, and attitude are propagated forward at a high rate in the mechanization using the IMU measurements. In both cases, with and without an IMU, the position and velocity states are in ECEF as opposed to a local frame. A more detailed discussion of the state propagation with IMU can be found in [35], [33], [43].

The receiver differential code bias states are treated as constants to be solved for. The STM is the identity matrix, and no process noise is added. The tropospheric wet delay estimate is treated as a random walk with a small amount of process noise. The code and carrier phase error terms are modeled as a first order Gauss-Markov processes that decays over time. These error terms help to absorb unmodeled errors, such as contributions from multipath or orbit and clock errors.

3.3.4 Measurement Models

The measurements used in the EKF are single and dual frequency code and carrier phase measurements along with single frequency Doppler measurements. Continuity
## 3.3. Filter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Process Noise Sigma</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock Bias</td>
<td>100</td>
<td>m</td>
</tr>
<tr>
<td>Forward Velocity</td>
<td>0.77</td>
<td>m/s</td>
</tr>
<tr>
<td>Vertical Velocity</td>
<td>0.11</td>
<td>m/s</td>
</tr>
<tr>
<td>Cross Track Velocity</td>
<td>1.18</td>
<td>m/s</td>
</tr>
<tr>
<td>Tropospheric Wet Delay</td>
<td>0.002/3600^{1/2}</td>
<td>m</td>
</tr>
<tr>
<td>Code Phase Multipath</td>
<td>0.2</td>
<td>m</td>
</tr>
<tr>
<td>Carrier phase error state</td>
<td>0.002</td>
<td>m</td>
</tr>
<tr>
<td>Float carrier phase ambiguity</td>
<td>0</td>
<td>m</td>
</tr>
<tr>
<td>Differential Code Bias</td>
<td>0</td>
<td>m</td>
</tr>
<tr>
<td>Frequency-Dependent DCB</td>
<td>0</td>
<td>m</td>
</tr>
</tbody>
</table>

Table 3.2: Process Noise Settings

of L1 carrier phase measurements is significantly better than that of the iono-free L1/L2 measurements used in the datasets in this chapter, which rely on semi-codeless L2 measurements. These semi-codeless L2P measurements are very prone to cycle slips and breaks in continuity, leading to the same breaks in continuity the dual frequency measurements. Using L1-only measurements in addition to dual frequency measurements leads to much smoother covariances and protection levels over time that do not jump up upon loss of L2 measurements. The use of L2C or L5 measurements would improve the continuity of the measurements due to their higher power but would decrease availability, as not all GPS satellites broadcast L2C or L5. The L1 measurements require ionospheric delay estimates. In this case, IGS TEC maps are used for both code and carrier measurements. However, the filter could be further simplified by only using L1 carrier phase measurements, not including an ionospheric estimate, and adding more process noise to the carrier phase error state due to the temporal changes in the ionospheric delay. L1 Doppler measurements are included for a more direct measurement of the receiver velocity. The code and carrier phase measurements are modeled as follows:

Dual frequency carrier phase:

\[
\Phi_{ij}^{(i)} = \| x_s^{(i)} - \hat{x}_{rx} \| + c(\hat{b}_{rx,c} - b_s^{(i)}) + m^{(i)} \Delta T^{(i)} + b_{pwu}^{(i)} - \hat{A}^{(i)} + R_m + \hat{M}^{(i)} + \hat{e}^{(i)} \tag{3.2}
\]
CHAPTER 3. PPP WITH INTEGRITY

Dual frequency code phase:

$$\rho_{\text{if}}^{(i)} = \|x_s^{(i)} - \hat{x}_{rx}\| + c(\hat{b}_{rx,c} - b_s^{(i)}) + m^{(i)}\Delta T^{(i)} - \hat{DCB}_{rx}^{(i)} - f_i\hat{FDCB}_{rx}^{(i)} + R_m + \hat{M}^{(i)} + \hat{\epsilon}^{(i)}$$  \hspace{1cm} (3.3)

Single frequency carrier phase:

$$\Phi_{\text{if}}^{(i)} = \|x_s^{(i)} - \hat{x}_{rx}\| + c(\hat{b}_{rx,c} - b_s^{(i)}) + m^{(i)}\Delta T^{(i)} + b_{\text{pwu}}^{(i)} - \hat{A}^{(i)} - I^{(i)} + R_m + \hat{M}^{(i)} + \hat{\epsilon}^{(i)}$$ \hspace{1cm} (3.4)

Single frequency code phase:

$$\rho_{\text{if}}^{(i)} = \|x_s^{(i)} - \hat{x}_{rx}\| + c(\hat{b}_{rx,c} - b_s^{(i)}) + m^{(i)}\Delta T^{(i)} - \hat{DCB}_{rx}^{(i)} - f_i\hat{FDCB}_{rx}^{(i)} + I^{(i)} + R_m + \hat{M}^{(i)} + \hat{\epsilon}^{(i)}$$ \hspace{1cm} (3.5)

where:

- $x_s^{(i)}$: satellite position provided by external precise orbit product
- $\hat{x}_{rx}$: estimated receiver position
- $\hat{b}_{rx,c}$: estimated receiver clock bias
- $b_s^{(i)}$: satellite clock offset provided by external precise product
- $m^{(i)}$: tropospheric mapping function
- $\Delta T^{(i)}$: estimated delta tropospheric delay
- $\hat{M}^{(i)}$: estimated multipath delay on the signal
- $\hat{DCB}_{rx}^{(i)}$: estimated receiver differential code bias per signal (shared across SVs)
- $f_i$: GLONASS signal frequency channel number from -7 to 6
- $\hat{FDCB}_{rx}^{(i)}$: estimated frequency-dependent GLONASS differential code bias per signal (shared across SVs)
- $I^{(i)}$: ionospheric delay/advance
- $R_m$: Other modeled effects. This includes relativistic effects, solid earth tide modeling, satellite antenna phase center offset and variation, ocean loading, modeled tropospheric delay, and any other desired range models. These are strictly modeled and not estimated.
- $\hat{\epsilon}^{(i)}$: other unaccounted for errors
3.3. FILTER DESCRIPTION

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Sigma</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pseudorange</td>
<td>2 [m]</td>
</tr>
<tr>
<td>Carrier phase</td>
<td>0.03 [m]</td>
</tr>
<tr>
<td>Doppler</td>
<td>0.05 [m/s]</td>
</tr>
</tbody>
</table>

Table 3.3: Measurement Noise Settings

Receiver differential code bias states have been included to accommodate the inclusion of single frequency measurements. For non-GLONASS constellations, one signal is assigned as the reference, and a constant DCB is estimated for every other signal. There is no process noise added to the DCB state. For GLONASS, a frequency-dependent DCB is included for each signal to account for local delays that are a function of the satellite frequency channel. We introduced time-varying state representing multipath error on code and carrier phase measurements, which was modeled as first order Gauss Markov process:

\[ M^{(i)}(t + \tau) = M^{(i)}(t)e^{\frac{\tau}{T_{mp}}} \]  

(3.6)

The time constant \((T_{mp})\) was set to 100 seconds per the multipath analysis in [49].

Adding additional measurements and states comes at the cost of computational complexity and run time, so steps have been taken to simplify the filter. One key place to reduce run time is in the measurement update of the covariance, which occurs for each subset. By decreasing the number of states and/or measurements, the measurement update speed can be increased. The first step is to not include L2-only measurements (iono-free measurements are still used, but single frequency L2 measurements are excluded), as they do not significantly improve accuracy but do increase run time. Second, a multipath state has been described for code phase measurements, and a similar error state should be included for carrier phase measurements so that the covariance properly characterizes the error. Such error states could capture multipath error, ionospheric delay error (for the single frequency case), or precise clock and ephemeris error. Rather than include one or more additional states into the carrier phase model, the error state is combined with the float carrier phase ambiguity state. In this case, a small amount of process noise is added. The
filter uses $0.0001^2 \text{ m}^2$ multiplied by the time step.

$$\hat{E}_{\text{carrier}}^{(i)} = \hat{M}_{\text{carrier}}^{(i)} + \hat{A}^{(i)} + \hat{\epsilon}_{\text{carrier}}^{(i)}$$  \hfill (3.7)

### 3.3.5 Measurement Update with Residual Checks

The measurement update step includes checks on the measurement residuals in order to exclude outliers. This is done by iteratively performing a measurement update and excluding measurements whose residuals exceed successively smaller thresholds until the desired small thresholds are reached. This is done because if the tight thresholds are used immediately, and measurements with large error are included, those measurements can drag the solution off sufficiently that otherwise good measurements can be excluded as well. In the case of carrier phase measurements, this means reinitializing the float carrier phase ambiguity estimates unnecessarily. As the filter converges, it is better able to detect erroneous measurements, and so the thresholds must be tightened. The exclusion of outliers here allows the fault rate at the point of the protection level computation to be kept lower than it otherwise would be, as simple things like missed cycle slips have a smaller impact on the fault rate because they are largely caught at the measurement update. The thresholds at their smallest are three times the measurement noise.

### 3.3.6 Inclusion of an IMU

When an IMU is incorporated into the solution, several changes are made. Tight coupling of IMU and GNSS has been well studied [64][43][33], so only a brief description relaying specifics of this implementation will follow in this section.

The addition of the IMU essentially replaces the position, velocity, and attitude propagation in the time update step of the Kalman filter as well as adds several new states and measurements in the measurement update step. The filter is an error-state filter implemented in the Earth centered Earth fixed (ECEF) frame. Euler angles are used to represent the attitude in ECEF [33].

The measurement update now helps to update IMU bias terms, where the IMU
3.3. FILTER DESCRIPTION

The model used is very simple and only estimates accelerometer and gyroscope bias. No scale factor or cross correlation terms are estimated for simplicity. Because this is not a perfect model of the IMU, additional process noise must be added to the bias terms in order to capture the unmodeled effects. The attitude of the IMU, and because we consider the vehicle to be a rigid body, the vehicle, is estimated. An important assumption made in this study is that the IMU is considered to be fault free, and all of the subset solutions use the same IMU measurements. Because no subset filters exclude the IMU measurements, an IMU fault would be undetected by the current setup.

Several additional pseudo-measurements are included with the IMU that significantly limit error growth by exploiting the physical nature of the automobile. The first constraint, implemented as a measurement, is the no-slip condition. Essentially, the velocity of the vehicle tires in the body frame cross-track direction must be zero. Non-zero velocity would indicate that the car is sliding, which is an extreme case that we are not considering. The second constraint is that the body frame vertical direction must be zero. Deviation from this constraint could arise from significant motion from the suspension of the vehicle or vehicle separation from the ground. The implementation of these constraints is described in Appendix B. Both of these constraints are considered to be fault-free for the purpose of this study, but this must be further studied in the future.

3.3.7 Computing Protection Levels

This section describes a number of different candidate protection level algorithms before finally evaluating each and selecting one in Section 3.4.2 that will be used for the following results. Many of the candidate protection level algorithms are based on work from [13].

The protection level (PL) is determined by:

1. the threat model
2. the nominal model
3. the test statistic
4. the integrity risk allocation
5. the upper bounds used to simplify the computation

The PL is defined such that:

\[
\text{Prob (position error} > PL \text{ & test passes)} \leq \text{Integrity risk allocation} \tag{3.8}
\]

Let us develop Equation 3.8 using the formula of total probability:

\[
P \left( \left| e^T (x - \hat{x}) \right| > PL \text{ & } y \in \Omega \right) = \sum_{i=0}^{N} P \left( \left| e^T (x - \hat{x}) \right| > PL, y \in \Omega | H_i \right) P \left( H_i \right) \tag{3.9}
\]

where:
- \( x \): the state
- \( \hat{x} \): all-in-view estimate of the state
- \( y \): the vector of measurements. For Kalman filter solutions, it refers to all measurements up to time \( t \), as well as the propagation equations
- \( \Omega \): the region (in the space of measurements) defined by the detection statistics
- \( e \): the vector that extracts the coordinate of interest from the state vector
- \( H_i \): the fault hypothesis

The probability \( P(H_i) \) is the prior probability of the fault mode. In this chapter we will assume a value of \( 10^{-5} \). This is the narrow fault probability used for GPS in ARAIM and only requires the monitoring of one-out subsets. Part of the future work will be the further investigation of the prior probabilities of fault in various environments. Section 3.4.6 also examines protection level availability if a \( P(H_i) \) of \( 10^{-4} \) is used.

The Kalman filter state estimate and covariance at time \( t \) coincides with the batch estimate that uses all previous measurements. As a consequence, the Kalman filter
3.3. FILTER DESCRIPTION

Protection levels can be directly derived from the snapshot case, both for solution separation and the residual based approach. The equations will be first introduced using the batch notation.

We will assume an optimal all-in-view estimator (in the sense of least squares). In this case, for every fault tolerant filter, we have the key property [18]:

\[
\sigma_{ss}^{(i)2} = \sigma^{(i)2} - \sigma^{(0)2}
\]  

(3.10)

where: \( \sigma^{(i)2} \): the variance of the estimation error of the subset filter \( i \) (the index \( i = 0 \) corresponds to the all-in-view)

This property is key because it allows us to easily update the solution separation standard deviation based on the subset filter and the all-in-view filter.

3.3.7.1 Detection Statistics: solution separation and sum of square residuals

As mentioned above, the protection level is a function of the detection statistics. We examine the two most common the detection statistics: solution separation and the sum of squared residuals.

3.3.7.1.1 Solution separation detection statistics  For solution separation, the detection region is defined by the difference between the fault tolerant subset solutions and the all-in-view solution:

\[
\Omega_{ss} = \{y | e^T (\hat{x} - \hat{x}^{(i)}) \leq T_i \}
\]  

(3.11)

where: \( \hat{x}^{(i)} \): a position that is tolerant to fault mode \( i \).

The threshold \( T_i \) is set to meet a predefined probability of false alert \( P_{fa} \) under nominal conditions.

\[
T_i = Q^{-1} (\alpha_i P_{fa}) \sigma_{ss}^{(i)}
\]  

(3.12)
where: $\sigma_{ss}^{(i)}$: the standard deviation of the solution separation under nominal conditions
$\alpha_i$: the allocation of the probability of false alert to the fault mode.
The sum over all modes must not exceed one.
$Q$: the complement of the normal CDF, and $Q^{-1}$ its inverse.

3.3.7.1.2 Sum of squared residuals statistic
The detection region for the sum of the square residuals is defined by:

$$\Omega_{\chi^2} = \left\{ y \mid y^T \left( W - WG (G^T WG)^{-1} G^T W \right) y \leq T_{\chi^2} \right\}$$  \hspace{1cm} (3.13)

where:
$G$: the batch observation matrix (and therefore including the propagation step)
$W$: the inverse of the measurement covariance

The threshold $T_{\chi^2}$ is set to meet the false alert, so it is defined by:

$$P\left( y \notin \Omega_{\chi^2} \right) = P\left\{ y^T \left( W - WG (G^T WG)^{-1} G^T W \right) y > T_{\chi^2} \right\} = 1 - F_{\chi^2}(T_{\chi^2}, n - p) = P_{FA}$$  \hspace{1cm} (3.14)

where: $F_{\chi^2}$: the CDF of a chi-square distribution with dof degrees of freedom
$n$: the number of measurements, and $p$ the number of unknowns.

As we will see in the next subsection, we use a recursive formula to update the sum of square residuals in the Kalman filter.

3.3.7.1.3 Recursive formula for the sum of squared residuals
For solution separation, the test statistic is formed using directly the outputs of the all-in-view filter and the subset solution filters. For the sum of squared residuals we need a recursive formula to update the sum of squared residuals statistic as the state is
propagated and measurements are added. The update can be written either as a function of the innovations or the residuals:

\[
\chi^2(t) = \chi^2(t-1) + (y_t - G_t\hat{x}_{t|t-1})^T (W_t - W_t G_t P_{t|t} G_t^T W_t) (y_t - G_t\hat{x}_{t|t-1}) \quad (3.15)
\]

\[
\chi^2(t) = \chi^2(t-1) + (y_t - G_t\hat{x}_{t|t})^T (W_t + W_t G_t P_{t|t-1} G_t^T W_t) (y_t - G_t\hat{x}_{t|t}) \quad (3.16)
\]

where: \(\chi^2(t)\) : the chi-square statistic of the residuals using all measurements up to time t (what would have been obtained with a bath process using all measurements up to time t)

\[P_{t|t}\] : posteriori error covariance

\[P_{t|t-1}\] : a priori error covariance

\[\hat{x}_{t|t-1}\] : state estimate at time t given observations up to t-1

\[\hat{x}_{t|t}\] : state estimate at time t given observations up to t

\[G_t\] : observation matrix at time t

\[W_t\] : inverse of the covariance of the observation noise

\[y_t\] : vector of measurements at time t

The proof for Equations and is in [34].

### 3.3.7.2 Protection levels for solution separation

The principle of solution separation is to run a bank of filters, where each filter is fault tolerant to a fault or set of faults. The fault detection statistic is the difference between each of these solutions and the all-in-view solution. Solution separation algorithms offer a clear link between the threat model, the test statistics, and the protection level. In addition to the solution separation PL proposed in [13], we implement three different PLs for each detection statistic:

- “Exact” PL uses a very tight upper bound and maximizes the integrity risk
over the fault size

- First upper bound uses an upper bound independent of the fault size (this is the one used in [18] and [2]

- Second upper bound is a simplification of the previous one

### 3.3.7.2.1 Integrity risk for one fault mode for solution separation

It can be shown [18] that a tight upper bound of the integrity risk for one fault mode is given by:

$$ P \left( |e^T (x - \hat{x})| \geq PL, y \in \Omega_{ss} | H_i \right) \leq \max_{\beta} Q \left( \beta - \frac{T_i}{\sigma_{ss}^{(i)}} \right) \left( Q \left( \frac{PL - \beta \sigma_{ss}^{(i)}}{\sigma^{(0)}} \right) \right) + Q \left( \frac{PL}{\sigma^{(0)}} \right) \tag{3.17} $$

The first term in the upper bound can be written as follows:

$$ F_{ss} \left( \frac{PL - T_i}{\sqrt{\sigma^{(0)} + \sigma_{ss}^{(i)2}}}, \frac{\sigma_{ss}^{(i)}}{\sigma^{(0)}} \right) = \max_{\beta} Q \left( \beta - \frac{T_i}{\sigma_{ss}^{(i)}} \right) \left( Q \left( \frac{PL - \beta \sigma_{ss}^{(i)}}{\sigma^{(0)}} \right) \right) \tag{3.18} $$

where: $F_{ss}$: function of two variables determined numerically [18].

### 3.3.7.2.2 Exact PL for solution separation

Going back to Equation 3.9 and using Equations 3.17 and 3.18, we obtain the following implicit definition of the PL:

$$ 2Q \left( \frac{PL_{SS,exact}}{\sigma^{(0)}} \right) + \sum_{i=0}^{N} F_{ss} \left( \frac{PL_{SS,exact} - T_i}{\sigma^{(i)}}, \frac{\sigma_{ss}^{(i)}}{\sigma^{(0)}} \right) P(H_i) = PHMI \tag{3.19} $$

This equation can be solved iteratively using a half interval search. Upper bounds for the search are defined in the next paragraphs, or can be found in [18], [2].
3.3. FILTER DESCRIPTION

3.3.7.2.3 First upper bound for solution separation  This upper bound is the one that was chosen to analyze Advanced RAIM performance in [2], and is the default proposed in [18]. We have the upper bound:

\[ F_{SS} \left( \frac{PL - T_i}{\sigma^{(i)}}, \frac{\sigma_{ss}^{(i)}}{\sigma^{(0)}} \right) \leq Q \left( \frac{PL - T_i}{\sigma^{(i)}} \right) \]  

(3.20)

We now replace it in equation 3.19, and obtain:

\[ 2Q \left( \frac{PL_{SS,approx1}}{\sigma^{(0)}} \right) + \sum_{i=0}^{N} Q \left( \frac{PL_{SS,approx1} - T_i}{\sigma^{(i)}} \right) P(H_i) = PHMI \]  

(3.21)

This equation is also solved used a half interval search but is much faster than 3.19, because we use the complement of the normal CDF \( Q \) instead of \( F_{SS} \).

3.3.7.2.4 Second upper bound for solution separation  This upper bound is defined by:

\[ PL_{SS,approx2} = \max \left( T_i + Q^{-1} \left( \frac{PHMI}{N \times P(H_i)} \right) \sigma^{(i)} \right) \]  

(3.22)

A complete proof of the integrity of this PL can be found in [18]. It is based on the inequality:

\[ Q \left( \frac{PL_i - T_i}{\sigma^{(i)}} \right) \leq \frac{PHMI}{N \times P(H_i)} \]  

(3.23)

3.3.7.2.5 Choice of thresholds  For a given \( P_{FA} \), any choice of the allocations that meets \( \sum \alpha_i = 1 \) (see Equation 3.12) is valid. The default chosen in [18], [2] is to allocate equally the PFA budget across the fault modes. It is possible to improve the PLs by optimizing this allocation. Although an optimal allocation (for the PL defined as in 3.21 ) is possible [13], it is computationally expensive. We use instead a suboptimal scheme which consists in making all the thresholds equal:

\[ \alpha = \frac{1}{N} \]  

(3.24)
3.3.7.3 Protection levels for the sum of squared residuals

As for the solution separation test statistic, we propose three different protection level equations, an exact one and two approximations. Before providing the protection level equations, we give the formula for the slope corresponding to each fault mode.

3.3.7.3.1 Slope formula The slope of a given mode is linked to the standard deviation of the solution separation statistic \[13],[44] through the equation:

\[
\text{slope}_i = \frac{\sigma_{ss}^{(i)}}{\sigma^{(0)}}
\] (3.25)

This means that it can be computed recursively using the subset filter covariance update via Equation 3.10.

For the sum of squared residuals, the integrity risk for one fault mode can be computed using the formula \[15]:

\[
P(|x - \hat{x}| \geq PL, y \in \Omega_{\chi^2} | H_i)
= \max_{\lambda} P_{ncx} (\chi^2, n - p, \lambda^2) \left( Q \left( \frac{PL - \lambda \sigma_{ss}^{(i)}}{\sigma^{(0)}} \right) + Q \left( \frac{PL + \lambda \sigma_{ss}^{(i)}}{\sigma^{(0)}} \right) \right)
\] (3.26)

where: \( P_{ncx} (K, dof, \lambda^2) \) : the non-central chi-square cdf with non-centrality parameter \( \lambda^2 \) and \( dof \) degrees of freedom.

We note:

\[
F_{\chi^2} \left( \frac{PL}{\sigma^{(0)}}, \frac{\sigma_{ss}^{(i)}}{\sigma^{(0)}}, \chi^2, n - p \right)
= \max_{\lambda} P_{ncx} (\chi^2, n - p, \lambda^2) \left( Q \left( \frac{PL - \lambda \sigma_{ss}^{(i)}}{\sigma^{(0)}} \right) + Q \left( \frac{PL + \lambda \sigma_{ss}^{(i)}}{\sigma^{(0)}} \right) \right)
\] (3.27)

3.3.7.3.2 Exact protection level for sum of squared residuals The “exact” protection level is computed using the integrity risk equation 3.9 and:
3.3. FILTER DESCRIPTION

\[ 2Q \left( \frac{PL_{\chi^2,exact}}{\sigma(0)} \right) + \sum_{i=0}^{N} F_{\chi^2} \left( \frac{PL_{\chi^2,exact}}{\sigma(0)}, \frac{\sigma_{ss}^{(i)}}{\sigma(0)}, T_{\chi^2}, n-p \right) P(H_i) = PHMI \quad (3.28) \]

As for the exact solution separation \( PL \), we use a half interval search.

3.3.7.3.3 First upper bound for the sum of squared residuals statistic

This upper bound is obtained by using the fact that the sum of squared residuals is an upper bound of the normalized solution separations [18]. It is defined by:

\[ 2Q \left( \frac{PL_{\chi^2,approx1}}{\sigma(0)} \right) + \sum_{i=0}^{N} Q \left( \frac{PL_{\chi^2,approx1} - \sigma_{ss}^{(i)} \sqrt{T_{\chi^2}}}{\sqrt{\sigma(0)^2 + \sigma_{ss}^{(i)2}}} \right) P(H_i) = PHMI \quad (3.29) \]

3.3.7.3.4 Second upper bound for the sum of squared residuals statistic

This coarser upper bound uses the same approach as Equation 3.22:

\[ PL_{\chi^2,approx2} = \max \left( \sigma_{ss}^{(i)} \sqrt{T_{\chi^2}} + \sqrt{\sigma(0)^2 + \sigma_{ss}^{(i)2}} Q^{-1} \left( \frac{PHMI}{N \ast P(H_i)} \right) \right) \quad (3.30) \]

3.3.7.3.5 Robustness against assumptions

It is important to note that the PLs defined by Equations 3.21, 3.22, 3.29, and 3.30 are more robust to the error model assumptions, in the sense that we only need the nominal error model to be an overbound (in the sense of [27],[71], or [14]) of the nominal errors.

This is not the case for the PLs defined by Equations 3.20 and 3.28 which rely on exact gaussian distributions, in particular on the fact that the test statistic is independent of the error estimate.
3.4 Results

3.4.1 Datasets

We used six data sets in this chapter, one comprised of a static receiver in open sky conditions, four comprised of a receiver on an automobile in various conditions, and one comprised of a receiver on an aircraft under flight conditions. In the following chapter, a much larger set of flight data will be evaluated.

The static data set was collected at the Stanford Aeronautics/Astronautics department rooftop by the receiver STFU (IGS MGEX Network designation):

- Trimble NetR9
- 1 hour of static data on November 7, 2017
- 1 Hz data
- GPS (L1 C/A-L2P semi-codeless), GLONASS (L1C-L2P)
- Truth position from IGS daily station solutions

Figure 3.4: Receiver antenna used to collect the static data at Stanford University

All of the automobile datasets use the same receiver configuration:
3.4. RESULTS

- NovAtel OEM 7500
- Tactical grade IMU operating at 100 Hz
- 1 Hour Driving Data on March 1, 2018
- GPS (L1 C/A - L2P semi-codeless), GLONASS (L1 C/A-L2P) at 1 Hz
- Truth positions provided by NovAtel OEM729 with tactical-grade IMU with forward and reverse processing

In each case, the lever arm between the GNSS antenna phase center and the IMU has been surveyed.

The open sky dynamic data, shown in Figure 3.5, set was taken just outside Calgary International Airport. The environment is fairly benign from a GNSS measurement standpoint, with only minor obstructions along the route.

![Figure 3.5: Open sky dynamic data set collected in Calgary](image)

The second dynamic data set, shown in Figure 3.6, travels through suburban Calgary. Large sections of the environment are fairly benign, but there are points where there are partial or full measurement outages.
CHAPTER 3. PPP WITH INTEGRITY

The next data set, shown in Figure 3.7, comes from a highway scenario. While the image in Figure 6 looks initially to be generally open sky, the large overhead traffic signs and overpasses in fact cause partial or full measurement outages quite frequently. This environment is much harsher than the suburban scenario because of these features and the frequency with which they are encountered due to the speed of the vehicle. This is also a one-hour data collection. Figure 3.7 shows that the typical speed throughout this hour is approximately 60 MPH.

The final data set consists of flight data from a roughly two hour flight near Philadelphia, Pennsylvania.

- Receiver: Trimble BD935
- 2 Hours Flight Data on June 30, 2017
- GPS (L1 C/A -L2P semi-codeless), GLONASS (L1 C/A-L2P) at 1 Hz
- Truth positions provided by the Natural Resources Canada Canadian Spatial Reference System Precise Point Positioning (CSRS-PPP) service
3.4. RESULTS

Precise clocks and orbits used in all data sets were from the Center for Orbit Determination in Europe (CODE) for all constellations (IGS MGEX product). IGS satellite antenna phase center offsets and variations were used. Ionospheric TEC maps were also from CODE.

Figure 3.7: Dynamic data collected in a highway environment

Figure 3.8: Dynamic flight data
3.4.2 Protection Level Algorithm Selection

In this section, the various candidate protection levels are compared under static open sky and dynamic open sky conditions. One of the protection level algorithms is then chosen for sole use in the following sections.

3.4.2.1 Parameter Settings

For these scenarios we used the following settings:

\[ PHMI = \frac{1}{3} 10^{-7} \]  
(3.31)

\[ P_{FA} = \frac{1}{3} 10^{-6} \]  
(3.32)

\[ P(H_1) = 10^{-5} \]  
(3.33)

The one third term accounts for all three coordinates. With these settings, only one-out subsets need to be formed. In Section 3.4.6 we examine results with two-out subsets as well.

3.4.2.2 Static Scenario Results

Figure 3.9: Position error and protection levels for the static scenario- on the right is a zoom in
3.4. RESULTS

3.4.2.3 Dynamic Scenario Results—open sky conditions

Figure 3.10: Position error and protection levels for the dynamic scenario—on the right is a zoom in

The spikes in the protection level (and to a lesser extent in the position error) are due to the loss of several measurements at the same location in the itinerary.

3.4.2.4 Remarks

In both scenarios, there were no detections (however, large outliers were excluded using a coarse threshold on the residuals, as indicated in Section 3.3.5). This means that for these scenarios, the actual measurement errors were compatible with the error models.

The protection levels achieved with solution separation are well below the ones achieved using the sum of squared residuals. This is expected because the solution separation statistic is, under some conditions, very close to the optimal detection statistics [15]. The very large gap between the two is mostly due to the very large threshold (and increasing with time) that is used for the sum of squared residuals. As we add degrees of freedom (number measurements minus number of new states), the threshold must increase to meet the probability of false alert. This is not the case for the solution separation statistic, which is modeled as a normal distribution.
(scaled by the standard deviation).

Another key point in these results is that the different solution separation PLs are very close to each other. As a consequence, there might be little value in implementing the exact PL as defined in 3.20, which is computationally intensive compared to 3.21 and 3.22. The approximate PLs do not require a worst-case bias search that increases the computational load significantly. This was not the case for the sum of squared residuals based protection levels, where the approximate protection levels were much larger than the exact search results.

The ”basic” protection level in Figure 3.10 is produced using the maximum test statistics (the difference between the subset solutions) and inflating that by the solution covariance of the subset. While this is a simple approach, it does not offer the fault detection capabilities of the other approaches. This is described in [51].

Ultimately, we chose the formulation of the PL given by Equation 3.22, as it offers PLs close to our best algorithm with a much lower complexity and computational load.

3.4.2.5 Speed

One of the drawbacks in solution separation is the need to run parallel filters. Figure 3.11 shows the run time as a function of the number of parallel filters for one hour of dual constellation data in our MATLAB implementation. For this plot we ran banks of filters two-out subsets in addition to the two-out subsets.

We observe that the run time for N filters (subsets) is much less than $N * \text{t}_{\text{one}}$ filter. For example, it takes less than 4 minutes to run 50 filters, as opposed to 50 times 2 min. This reduction (with respect to a na"ive implementation) was achieved using by sharing modelled propagation effects across subsets, as described in section 3.3.

3.4.3 Static Open Sky Scenario

For this and all following sections, the protection levels are computed using Equation 3.22. In this section, we used the static open sky data, and the results are shown in Figure 3.12. The filter is run in a dynamic mode just as it is for the automobile
3.4. RESULTS

Figure 3.11: Run time as a function of the number of subsets scenarios. That is, process noise is added to the position and velocity states at each time step. By ten minutes into the run, the protection levels converge to less than two meters in the East and North directions, and by the sixty minute mark, they are both under one meter.

Figure 3.12: Position errors and PLs in static scenario- on the right is a zoom in
3.4.4 Fault Injection Scenarios

In order to test the implementation of the filter and the PL, we inject faults into our measurements. The first fault is a step error of 20 meters added to the precise clock input for GPS PRN 8. This error is immediately caught by the initial residual checks in the measurement update step of the EKF.

![PRN 8 Measurement Acceptance](image)

Figure 3.13: PRN 8 Measurement Usage with Injected Step Fault

![Protection Levels with Injected Step Fault](image)

Figure 3.14: Protection Levels with Injected Step Fault
Whether or not the code and carrier phase measurements for PRN 8 are included in the measurement update of the EKF is shown in Figure 3.13. As soon as the fault is introduced, both the code and carrier phase measurements are excluded. The code phase measurements are not reintroduced, but the carrier phase measurements actually return to the filter with an offset carrier phase ambiguity estimate. The position error and protection levels are only slightly impacted due to weakened geometry and a carrier phase ambiguity estimate reset. In cases that are harder for the residual checks to detect errors, such as when the error slowly ramps, the method of fault detection using solution separation is still capable. If a protection level threshold is tripped, the exclusion algorithm is triggered. The exclusion algorithm compares the solutions of each subset to one another, and the subset that is furthest from the others is determined to be the fault-free subset. The that subset state becomes the new all-in-view state, the covariance is re-initialized, and what is believed to be the faulted PRN is excluded from all subsets from then on.

Figure 3.15: Injected Ramp Fault in Static Scenario
Figure 3.15 illustrates this process. A ramp error was injected into PRN 8 precise clock- 9 meters per hour. This ramp was slow enough that much of the ramp was pulled into the error states associated with PRN 8. Once the PL threshold is tripped, all PRN 8 measurements are excluded henceforth, and the filter is reset.

### 3.4.5 Dynamic Open Sky Scenario

For these plots, we use the open sky automobile data set, which is the open sky driving scenario. This is dynamic data under generally favorable conditions. Dropouts of measurements on the L2 frequency occur often, but L1 is never completely lost, leading to smooth errors and protection levels, which converge to less than 1.5 meters in the East and North directions.

![Graphs showing position errors and PLs for dynamic scenario in open sky](image)

Figure 3.16: Position errors and PLs for dynamic scenario in open sky- on the right is a zoom in

### 3.4.6 Dynamic results in open sky conditions with two out

In Section 3.3.7, a probability of fault of $10^{-5}$ was introduced. However, larger values may be necessary based on the environment and error checking capabilities of the
3.4. RESULTS

This section investigates the impact on the protection levels if larger fault probabilities are used. For this run, we assumed a probability of fault large enough ($10^{-4}$) so that two-out subsets had to be considered as well. The protection levels do increase compared to the case where only one out subsets are considered (Figure 3.17) but not very significantly. This is due to the fact that the geometry is very strong, so that the subset geometries are only marginally worse than the all-in-view.

![Figure 3.17: Position errors and PL for dynamic scenario in open sky conditions with two satellites out- on the right is a zoom in](image)

3.4.7 Suburban Scenario

The environment is generally fairly benign and close to open sky, but there are several moments during the collection where partial or full GNSS measurement outages are experienced. Figure 3.6 shows one of the streets driven in the one-hour data collection. It also shows the speed of the vehicle throughout the collection, which is generally fairly slow and remains mostly in the 10 to 30 MPH range.

Figure 3.18a shows the position error and protection levels produced for the GNSS-only solution in the suburban scenario. The protection level, driven by the filter covariances, converges over the first 10 or so minutes to the 1-2 meter range in the East
and North directions and to approximately 4 meters in the vertical direction. These protection levels bound the error quite well. However, one feature that stands out from this figure is that the protection levels jump and must reconverge several times throughout the single hour. This is driven by the partial or full GNSS measurement outages as the vehicle passes under tree cover or next to tall buildings. As signals are occluded and measurements are missing, the covariance swells. The increase in the position covariance is, as is typical for an EKF, driven by the process noise added to the position and velocity states. Without a particular dynamics model or high update rate, the process noise added must be able to account for any dynamics the automobile may go through during the interval between a measurement update and the subsequent time update. When the signals are available later, the carrier phase ambiguities must be re-estimated, which takes time to converge, leading to the characteristic PPP convergence multiple times throughout the data collection.

![Figure 3.18: Position errors and PLs for suburban dynamic scenario](image)

(a) PPP Only  
(b) PPP+IMU

Figure 3.18: Position errors and PLs for suburban dynamic scenario

The inclusion of a tightly-coupled IMU to the PPP solution allows for the short gaps in measurements to be bridged and the protection levels to be kept down, as shown in Figure 3.18b. As before, the protection level must converge upon initialization, but the protection levels do not exhibit the same jumps as they did in the GNSS-only case. The IMU and the land vehicle constraints are able to limit the growth of the position and velocity covariance over the short measurement outages
3.4. RESULTS

enough to almost entirely smooth out the protection levels. The limited covariance growth allows for the carrier phase ambiguities to be re-estimated with fairly tight confidences upon signal reacquisition.

This growth suppression is, of course, accompanied by an improvement in solution accuracy, especially just following the outages. The protection levels converge to the 1-2 meter range in the horizontal directions and maintain that through the one hour duration. There are sections when the protection level does increase somewhat, and this is driven by worsened GNSS geometry, which does still drive the protection levels long term.

3.4.8 Highway Scenario

While the image in Figure 3.7 looks initially to be generally open sky, the large overhead traffic signs and overpasses in fact cause partial or full measurement outages quite frequently. This environment is much harsher than the suburban scenario because of these features and the frequency with which they are encountered due to the speed of the vehicle. This is also a one-hour data collection. Figure 3.7 also shows that the typical speed throughout this hour is approximately 60 MPH.

Figure 3.19a clearly shows the effects that the various overhead obstacles have on the ability for the filters to converge. Approximately once per minute, enough measurements are occluded that the covariances, and thus the protection levels, experience a jump. In the East and North directions, the measurement continuity is so poor the protection levels never reach two meters. In the vertical direction, the protection level never reached five meters and is more typically approximately 10 meters.

As before, the impact of the addition of the IMU to the state propagation is clear in Figure 3.19b. The protection levels do not experience consistent spikes, and the accuracy is greatly improved. However, due to poorer overall geometry, the protection levels are slightly larger than in the suburban case, typically in the 3-4 meter range in the East and North directions. Still, this is a dramatic improvement when compared to the GNSS-only solution.
This chapter has dealt with a MEMS tactical grade IMU, which could cost thousands of dollars, likely pushing them out of the budget for even an autonomous car. However, less expensive IMUs can offer acceptable performance but cannot coast for as long. In Appendix B, long-GNSS-outage scenarios are investigated using the tactical grade IMU, but in this chapter, the GNSS outages never last more than one or two seconds. In only short-outage cases (<2 seconds), even automobile grade IMUs costing approximately $100 can suffice for protection levels down to 1 meter \[77\] when vehicle dynamic constraints are taken into account. With odometry measurements, acceptable protection levels during GNSS outage can be available for even longer.

![Figure 3.19: Position errors and PLs for highway dynamic scenario](image)

(a) PPP Only

(b) PPP+IMU

Figure 3.19: Position errors and PLs for highway dynamic scenario

### 3.5 Conclusion

We have formulated RAIM protection level formulas using either solution separation or the sum of residual squares. Both formulations consist on straightforward adaptations of snapshot RAIM to a Kalman filter solution. For solution separation, we have shown an implementation where the computational cost of running a bank of filters is far from being proportional to the cost of one filter. Instead, we could run 50 additional filters for the cost of one. For residual based RAIM we have developed a set of formulas to update the sum of square residuals from one time step to the next one. Because this test statistic is exactly the same as the one used in snapshot
3.5. **CONCLUSION**

RAIM (when we consider the problem as a batch least squares), we could use the formula that ties the slope of a fault mode to the standard deviation of the solution separation. The slope can therefore also be updated recursively. Solution separation, even in its simpler form, outperforms the residual based approach developed here both in performance and computational cost:

- exact PLs based on the residual based approach are two to three times larger than the coarser upper bound for solution separation.
- computing the slope for residual based is equivalent to updating the subset covariance matrix
- the exact PL for RB was found to be computationally expensive, as it requires a numerical search of the worst case bias for each fault mode
- the approximate PLs for solution separation do not require a worst case bias search and are very close to the exact PL.

Residual based approaches could still be valuable if the performance they offer is sufficient. However, it is unlikely that the computation of the slope will be significantly lower than the cost associated to the subset Kalman filter. This is due to the fact that the slope computation involves the inversion of a matrix that is of the same size as the subset covariance (and actually very closely related), and it is this inversion that dominates the computational load. We examined the effect of considering multiple faults in the formulation of the test statistics and the protection levels. The results are very promising: protection levels below 2 m appear to be achievable, and the computation load is lower than expected. Finally, the inclusion of the IMU was shown to improve the continuity and availability of low protection levels when compared to the GNSS-only solution, and 1-2 meter protection levels have been produced in the horizontal directions. In order to reduce the computational load of the solution separation system, the grouping of fault modes (subsets) is also used, which, with the IMU only leads to a small performance impact to the protection levels.
Chapter 4

Integrity with Broadcast Navigation Data

4.1 Introduction

This chapter extends the estimator designed for PPP described in the previous chapter to use GPS broadcast navigation messages and SBAS corrections in conjunction with integrity algorithms originally developed for ARAIM to produce protection levels of less than 10 meters.

PPP offers high accuracy, global positioning, and there is growing enthusiasm for the application of PPP techniques to safety critical systems [60], [22]. We have shown that PPP, in conjunction with techniques developed for integrity in aviation, can be used to produce meter-level protection levels for static, automotive, and flight scenarios [34]. However, PPP requires real-time, precise orbit and clock corrections, which may not always be available. There have been explorations into using SBAS corrections or broadcast navigation messages for PPP [70], [42], [88], but these have been focused on accuracy rather than integrity. Using SBAS corrections with dual-frequency PPP algorithms, decimeter-level accuracy has been found after convergence. The goal of this chapter is to develop and analyze the use of SBAS orbit and clock corrections or simply the broadcast navigation messages with a PPP engine and an integrity algorithm based on solution separation like that used in Advanced RAIM. While SBAS
using traditional processing techniques can produce protection levels on the order of
tens of meters, it is possible PPP techniques can reduce these protection levels.

The PPP algorithm used is based on a simple EKF that estimates position,
clock, troposphere, float carrier phase ambiguity, and error states. Solution separation
requires that multiple filters are run, each of which is tolerant to a fault or set of faults.
The number of subsets, i.e. additional filters, is determined by the probability of each
fault mode. Solution separation also requires the careful characterization of the error
sources so that the nominal covariance produced by the EKF conservatively describes
the actual error. One of the goals of this chapter will be to analyze the evolution of the
orbit and clock error for both SBAS and the broadcast navigation messages so that
the covariance matches the true error. In particular, states in the PPP filter estimate
the error on that signal, which would take into account the orbit and clock error.
Navigation message handover can produce discontinuities in that error, but with the
knowledge of when the handover takes place and by staggering the handovers per
satellite, the error can be mitigated.

Tests are run for both static and aviation scenarios, using orbit and clock corrections
of three varieties: precise orbit and clock products from IGS analysis centers;
WAAS SBAS orbit and clock corrections; and broadcast navigation message orbit
and clock estimates. Position estimates and protection levels are produced from each
case, and these can be compared to non-PPP WAAS solutions and protection levels.
Dual frequency measurements are used. For each of these cases, the nominal error
characterization and probability of fault will be assessed. The static data source is
a Trimble NetR9 on the roof of the Stanford Aeronautics/Astronautics building in
California, USA. The aviation data source is another receiver aboard a Global 5000
aircraft that is owned and operated by the FAA Technical Center in New Jersey,
USA.

Finally, this chapter introduces methods for fault detection and exclusion using
solution separation and sequential filters that maintains low protection levels through
the fault event.
4.1.1 Prior Art

Much of the prior art listed in Section 3.2 is also valid for this chapter, but a few works will be highlighted in this section. In particular, the use of SBAS data as PPP corrections has been explored. Rho [70] incorporated SBAS corrections into a sequential PPP filter and found improvements in ranging accuracy over the standard broadcast navigation message. Hesselbarth [42] further expands this area of study to include European and Japanese augmentation systems. However, these works are primarily interested in the use of SBAS to improve accuracy but not integrity. On the topic of integrity, Joerger [46] introduces the use of the broadcast navigation message with ARAIM and sequential filters, which this chapter builds off of, but the work in [46] uses residual-based protection levels.

4.2 Methods

4.2.1 Estimator Design

The PPP algorithm with solution separation is implemented using an extended Kalman filter using dual frequency code and carrier phase measurements. Many of the details of the implementation can be found in Chapter 3. The states estimated are carefully chosen so as to leverage the structure of the problem. The predicted dual frequency code and carrier phase measurements can be modeled as follows:

Dual frequency carrier phase:

\[ \Phi_{if}^{(i)} = \|\hat{x}_s^{(i)} - \hat{x}_{rx}\| + c(\hat{b}_{rx,c} - b_s^{(i)}) + m^{(i)}\hat{T}^{(i)} + b_{pnuu}^{(i)} - \hat{A}^{(i)} + R_m + \hat{M}^{(i)} + \epsilon_{brdc}^{(i)} + \epsilon^{(i)} \]  \hspace{1cm} (4.1)

Dual frequency code phase:

\[ \rho_{if}^{(i)} = \|\hat{x}_s^{(i)} - \hat{x}_{rx}\| + c(\hat{b}_{rx,c} - b_s^{(i)}) + m^{(i)}\hat{T}^{(i)} - \hat{D}\hat{C}B_{rx}^{(i)} - f_i F \hat{D}\hat{C}B_{rx}^{(i)} + R_m + \hat{M}^{(i)} + \epsilon_{brdc}^{(i)} + \epsilon^{(i)} \]  \hspace{1cm} (4.2)

where \( \epsilon_{brdc}^{(i)} \) is the error due to broadcast navigation message orbit and clock. All of the other terms are described in section 3.3.4.
The estimated states are indicated by a carrot over the symbols. Here, the estimated states include the position, velocity, receiver clock biases, tropospheric delay, carrier phase ambiguities, multipath error, receiver differential code bias, and broadcast orbit and clock error. This model is mostly typical of a PPP implementation with one significant exception—the state tracking the error contribution of the broadcast orbit and clock on each line of sight. The error contributed by the broadcast orbit and clock can be handled by the filter in one of two ways. The first way is how it is typically handled, where the sigma associated with the navigation message is simply lumped into the measurement sigma. This is generally acceptable for snapshot positioning, but one part of the Kalman filter assumptions is that the measurement error is zero-mean and uncorrelated from epoch to epoch. If we include the broadcast orbit and clock in the not-estimated measurement error term, this assumption will not hold. Figure 4.1 shows the error in the GPS broadcast navigation message orbit and clock estimates as projected onto a terrestrial user’s line of sight. One can see from this figure that the error varies slowly over the course of the day, with jumps only occurring on the changeover from one navigation message to the next. We can leverage this correlation in our estimation of the orbit and clock error by including it as a state for each line of sight and only adding enough process noise to capture the slowly changing error. A characterization of the rate of change of the error in the broadcast orbit and clock is shown later in this chapter.

4.2.2 Integrity Algorithm Design

We use solution separation-based protection levels derived for Kalman filter navigation solutions as described in Chapter 3. The use of solution separation requires that banks of Kalman filters are run in parallel, each one tolerant to a fault or a set of faults. Much care has been taken to ensure that such banks are not overly computationally demanding, where the main step taken to reduce computational load is the sharing of the computation of modeling across all of the subsets. That is, when the orbit and clock are propagated to the desired time, this is only done once using the all-in-view solution as input, and the orbit and clock state are shared across all
subsets. This can be done because in general, these models are not sensitive to the meter or sub-meter level position differences between the subsets. Given the position estimates and covariances output by the bank of Kalman filters, protection levels are computed using algorithms originally developed for ARAIM which have been since modified for sequential filters. Ultimately, the protection levels are computed using equation 3.22.

The use of such algorithms requires careful characterization of the environment and the measurements used so that the nominal covariance faithfully reflects the actual error and the fault rates assumed similarly reflect the observed fault rates. The narrow fault probability assumed so far is $10^{-5}$, which means that only one-out subsets need be considered.
4.3 GPS Performance Characterization

The process noise that needs to be added to the clock and ephemeris error states can be estimated through an evaluation of the change in the actual error in the broadcast clock and ephemeris data as projected onto user LOS’s. In order to do this, GPS broadcast clock and orbit is compared to precise orbit and clock estimates provided by the National Geospatial-Intelligence Agency (NGA) and the Center for Orbit Determination European (CODE). This process has been described in [82] [30] [57] and is additionally described in Appendix A. The difference between the broadcast and precise estimates is computed at 5 second intervals using GPS clock estimates from CODE, and these differences in the Earth Centered Earth Fixed (ECEF) frame are projected onto the lines of sight of 200 evenly spaced global users over a period from January 1, 2018 to February 28, 2018.

The projected error is called the Signal in Space Ranging Error (SISRE). The difference from epoch to epoch of the SISRE at each user location is taken and made into a histogram, taking into account boundaries when there is an issue of data (IOD) handover and binning those samples separately.

The result of this process is shown in Figure 4.2. The changes in SISRE over time are very different within a navigation message and between consecutive navigation messages. For the core distribution in the blue, which represents the change in SISRE per second when the navigation message IOD does not change, can be overbounded by a normal distribution with a sigma value of 0.002 m to the $10^{-5}$ level. If one were to include the IOD crossings, one would need a one-sigma value of 0.16 m, so it is very important to separate the two cases. In fact, because one is able to control when the IOD changeover occurs, we simply reinitialize the estimate of the orbit and clock error when an IOD changeover happens, where the initial uncertainty is the broadcast URA value. This 0.002 m value matches very closely to the GPS SPS guaranteed accuracy for User Range Rate Error (URRE) at the 95% level. The result of this analysis is that the process noise added to the orbit and clock error state is $0.002^2$ m$^2$/s.

It is very common for multiple satellites to switch to the next navigation message
at the same time at two hour intervals. If we were to reinitialize all of the orbit and clock error estimates at the same time, the filter would essentially go through a soft reset. In order to mitigate this effect, we choose to stagger the changeover from one IOD to the next across the satellites. That is, when new navigation messages are available, only one new navigation message can be used at a time, and the rest of the satellites must use their older, but still valid, navigation messages. After a certain amount of time, which is set to 10 seconds, another PRN can next switch to a new navigation message. This reduces the impact of the added process noise needed to change from one navigation message to the next. Figure 4.3 depicts this process, where each dashed line indicates the IODE of a different GPS satellite. One satellite performs an IODE handover at the 10 second mark, another at 20 seconds, and so on.
4.3. GPS PERFORMANCE CHARACTERIZATION

4.3.1 Inclusion of WAAS Corrections

WAAS corrections are not directly applied to either the pseudorange or the carrier phase measurements for reasons that can be seen in Figure 4.4, which shows the line of sight orbit and clock corrections for an arbitrary user and PRN over a fifteen-minute period. The range-rate corrections in particular make it such that, if the corrections were directly applied to the pseudorange and code measurements, a large amount of process noise would need to be added into the broadcast error state at each epoch, which would lead to increased uncertainty in the overall covariance. Even without the range-rate corrections, the quantization of the corrections would lead to a similar effect, and because the corrections come at such a high rate, staggering the correction changeover as is done for the broadcast ephemeris becomes problematic. So, instead, the corrections are inserted into the filter as direct measurements of the line of sight broadcast orbit and clock error. Equations 4.3 shows the simple measurement model, where $\delta r_{WAAS}$ is used as a direct measurement of the signal in space ranging error. Equation 4.4 shows that the measurement noise used is provided by $\sigma_{FLT}$, which is
here the $\sigma_{UDRE}$ inflated by the covariance projection onto the specific line of sight, where the covariance is provided in WAAS Message Type 28. This measurement model is likely slightly optimistic, as it assumes that the measurement error on the corrections conforms to the Kalman filter assumptions that they are zero-mean and uncorrelated over time. As the WAAS orbit and clock estimator is a “faster” system than that of the GPS control segment because it does not have global reference stations and thus must quickly make adjustments to the estimates based on real time measurements, these assumptions may not be very far off. However, this will need to be further investigated in future works.

Measurement model:

$$\hat{\epsilon}_{brdc}^{(i)} = \delta r_{WAAS}$$  (4.3)

Measurement noise:

$$\hat{\sigma}_{\delta r_{WAAS}}^{(i)} = \sigma_{FLT}$$  (4.4)

Figure 4.4: WAAS Broadcast Orbit and Clock Corrections
4.4 Results

Several sets of results are provided: a nominal static scenario, and a number of flight scenarios. The static scenario and one of the flight scenarios are dual frequency code and carrier with GPS only; the rest of the flight scenarios also include Galileo measurements. In the first two scenarios, all subsets are initialized using the same state and covariance, representing a trusted, fault-free initial state estimate across all subsets. In the dual-constellation data, the filter has been updated such that each subset is initialized using a code phase solutions using the appropriate fault-tolerant subsets. This second configuration more closely resembles what would be fielded operationally.

4.4.1 Static Scenario

The first set of results comes from a Trimble NetR9 on the roof of the Aeronautics/Astronautics building at Stanford University. It is 3 hours of 1 Hz data on GPS L1 C/A and semi-codeless L2P. As this receiver is a member of the International GNSS Service’s (IGS) Multi-GNSS Experiment (MGEX) network, a daily high accuracy solution is produced. The daily MGEX solution is used as the truth reference in this section.

Figure 4.5 shows the position error and protection levels found using only the GPS broadcast navigation message, using WAAS corrections in addition to the broadcast navigation message, and using full PPP implementation. As is typical of PPP techniques that rely on estimating float carrier phase ambiguities, there is a convergence period in the protection levels for all three cases. First discussing the case using the broadcast navigation message without WAAS corrections, the convergence is very slow. The protection levels do not seem to reach any sort of steady state until after two hours have passed, due to the additional process noise and overall uncertainty provided by the navigation message error augmented states. However, within 15 minutes, the protection levels in the East and North directions are under 10 meters, and the vertical protection level is under 15 meters. After two hours of convergence, the protection levels in the East, North, and up directions are approximately 6 meters,
3.5 meters, and 7 meters, respectively.

The protection levels produced using WAAS corrections with the broadcast navigation message are close to those produced using the full PPP implementation with external corrections. Convergence occurs in the first 30 minutes, and the WAAS protection levels are generally within one meter of the PPP protection levels, at approximately 3 meters in the East and North directions, and 4 meters in the up direction. As previously mentioned, these protection levels use an optimistic measurement model. However, the accuracy is drastically increased using WAAS corrections to better than 15 centimeters RMS horizontal, and the position error normalized by the nominal covariance output by the all-in-view (AIV) filter remains very conservative. Finally, the error also does not exceed a small fraction of the protection level at any point during the run.

![Figure 4.5: Position error and protection levels using broadcast navigation data](image)
4.4.2 GPS-Only Flight Data

This section describes results from flight data collected on an FAA Global 5000 aircraft. The data shown in Figure 4.6 consists of flight data from a roughly two-hour flight near Philadelphia, Pennsylvania on June 30, 2017. The receiver used was a Trimble BD935, and the signals used were GPS L1 C/A and L2P semi-codeless. The truth against which the position estimates were compared were provided by the Natural Resources Canada Reference System Precise Point Positioning (CSRS-PPP) service.

Figure 4.6 shows the position error and protection levels derived for this scenario using only the GPS broadcast orbit and clock from the navigation message. As with the static data, there is a convergence period, where this period is longest for the broadcast navigation message data and under 30 minutes for the WAAS correction and PPP cases. The WAAS correction protection levels again rival those of the full PPP case. The broadcast-only protection levels are consistently less than 15 meters in every direction, which is a significant improvement over traditional RAIM or even SBAS protection levels, which are typically in the tens of meter range.

4.4.3 GPS-GAL Flight Data

Flight data collected over a period of approximately one year was used to evaluate ARAIM performance through momentary outages and cycle slips and due to aircraft dynamics. A multi-constellation, multi-frequency receiver (Trimble BX935-INS) tracked GPS, (L1 C/A and L5) and Galileo (E1 and E5a). This receiver was installed in a Global 5000 jet owned and operated by the William J. Hughes FAA Technical Center. It records and stores GNSS measurements whenever flights are taken [66]. The data collection effort of this section included data recorded over approximately 35 flights from September 2017 to April 2018.
Figure 4.6: Position error and protection levels for GPS-only flight data

Figure 4.7: Ground track and satellite visibility for GPS-Galileo example flight data

Figure 4.7 shows the trajectory and satellite availability of a data collection flight taken on September 20, 2017 (example number 6 in Figure 4.9). The aircraft takes off approximately 30 minutes into the collection and is in the air for two hours. At
least nine GPS and three Galileo satellites are in view at all times, but there are occasional brief breaks in measurement continuity. Figure 4.8 shows protection levels and position error computed for this flight using sequential solution separation and the broadcast navigation messages as well as snapshot ARAIM [18] for a baseline comparison. The protection levels start off at similar levels, but the sequential filters quickly converge to much tighter values. In addition to producing generally lower protection levels, the sequential approach offers robustness to cycle slips or other brief satellite outages. The snapshot ARAIM solution suffers jumps due to geometry changes or cycle slips, while the protection levels produced using the sequential approach remains smooth and stable. The protection levels shown in Figure 4.8 are larger than those in Figures 4.5 and 4.6 due to the differences in initialization mentioned in Section 4.4.

![Figure 4.8: Horizontal (left) and vertical (right) position error and protection levels for GPS-Galileo example flight data comparing snapshot ARAIM and sequential filters using only broadcast navigation data](image)

The mean protection levels across all of the flights are shown in Figure 4.9. The sequential approach offers protection levels between 30% and 70% smaller than those available through snapshot ARAIM.
CHAPTER 4. INTEGRITY WITH BROADCAST NAVIGATION DATA

4.5 Fault Detection and Exclusion

The following section addresses the detection and exclusion of faults and introduces a new method of FDE using solution separation with sequential filters that maintains low PLs through the FDE. A fault is injected into the static scenario in order to check that the fault detection and exclusion abilities of the system operate correctly. Two different methods are employed to compute the subsets and associated protection levels after fault exclusion. The injected fault is modeled after an observed GPS fault from 2010. The fault occurred on PRN 30 on February 22, 2010 and is shown in Figure 4.10 and Figure 4.10b. Figure 4.10 shows the broadcast clock and ephemeris error split into the radial, along-track, cross-track, and clock components as well as the maximum projected error (MPE), which is the worst-case error experienced by a terrestrial user. A fault begins at approximately 21:00 (indicated by the red shaded area), and the fault is caused by a clock runoff. This can be seen in Figure 4.10b, where the broadcast clock bias (red circles) and the true clock bias (blue dots) start to diverge as the onboard clock experiences a frequency offset leading to a ramp error. This fault can be modeled as a ramp of approximately 50 meters/hour. The fault is
4.5. FAULT DETECTION AND EXCLUSION

(a) Observed GPS Fault from 2010
(b) Precise and broadcast clock bias from PRN 30 fault

Figure 4.10: Injected Ramp Fault in Static Scenario

injected into the static scenario data by adding such a ramp to all measurements from a single PRN starting 20 minutes into the run.

The following section describes two methods for reinitializing the PPP filters once a fault has been detected and excluded. The original method, as used in Chapter 3, requires that the entire filter, including the subset filters, be completely restarted, leading to a sizable jump in the protection levels. The new method to be described relies on carrying extra subset filters that are not used in the protection level computations and are only used in the FDE process. The two methods will be described here.

4.5.1 Original method for FDE

The first method involves a simple reinitialization of the AIV and subset filters after fault detection and exclusion. The only subsets that are computed are those required to produce protection levels at the current epoch. The left part of Figure 4.11 shows a diagram of measurement usage for the all in view (top row) and the other subsets (rows 2 onward). In the diagram, each row is a different subset, and each column is a different PRN, where the filled blue circles indicate that the measurement is used, and the empty circles indicate that the measurement is excluded. For solution separation
considering only one-out subsets, all that is needed to produce a protection level is the subsets included in the left diagram.

Figure 4.11: Measurement usage diagram for original fault detection and exclusion method

Figure 4.12: Position error and protection levels given injected ramp fault and original FDE method

Figure 4.12 shows the results of using this method given an injected fault as described previously. Twenty minutes into the run, the ramp fault begins, which can be seen in the growth in the position error. Just before the thirty-minute mark, the growth in the solution separations exceeds the computed threshold, indicating that a
fault has been detected. The fault, on PRN 2, is identified, and the new desired AIV and subsets must have PRN 2 excluded. This is shown on the right side of Figure 4.11. The new AIV is actually a subset of the original set, but because none of the new subsets have been tracked previously, the already tracked AIV cannot be used, and the entire filter must be reinitialized. The reason that the previously tracked AIV cannot be used is that if new subsets were spawned given an already converged AIV, those subsets would not be tolerant to faults on measurements that have already been included in the AIV solution. Once the filters are reinitialized, the protection level spikes, and because of the weakened geometry due the missing PRN, the protection levels take longer than usual to converge again.

### 4.5.2 New method for FDE with “on-deck” filters

![Measurement usage diagram for “on-deck” fault detection and exclusion method](image)

In order to avoid the large jump in protection levels upon reinitializing the filters after FDE, a new process has been developed wherein “on-deck” subsets are produced and are only used in case of FDE. Essentially, if we are only concerned with one-out subsets as we have been, then the on-deck subsets would be the two-out subsets, as shown in the left side of Figure 4.13. These subsets are not included in the protection level computations.

Figure 4.14 shows the results given the injected fault and the new FDE method.
Figure 4.14: Position error and protection levels given injected ramp fault and "on-deck" FDE method

involving the on-deck subsets. At the same point as in Figure 4.12, a fault is detected. However, the filters do not need to be reinitialized because the new AIV and subsets have been tracked all along. These filters are highlighted by the red boxes on the right of Figure 4.13. The new on-deck filters need to be reinitialized at this point, but as long as another fault is not detected before those filters converge, this does not impact the protection levels. The new AIV was one of the original subsets filters, and the new primary subsets were already converged in the on-deck filters. It is, of course, computationally expensive to carry around the on-deck filters, but techniques such as grouping of the subsets have been explored [12], [16], which can dramatically reduce the computational load. After FDE, the protection levels are slightly increased due to the weakened geometry, but most importantly, they are already converged.
4.6 Conclusion

A system has been developed to produce horizontal protection levels of less than 15 meters after convergence using only broadcast navigation messages and external inputs. The navigation algorithm relies on an extended Kalman filter and with states that track the error contribution of the broadcast orbit and clock. The integrity algorithm uses solution separation with protection levels similar to those used for ARAIM. With WAAS corrections, protection levels rivaling those of a full PPP system have been found, and broadcast-only protection levels have been found that are far lower than those produced by SBAS or snapshot ARAIM. Finally, a new method for maintaining converged protection levels through FDE for EKF-based solution separation has been introduced. These methods provide an opportunity for protection levels 30% to 70% lower than have been previously shown given only data broadcast on open satellite signals.
Chapter 5

Constellation Monitoring

5.1 Introduction

The use of a GNSS in Advanced RAIM requires the careful, long-term characterization of its signal-in-space (SIS) performance. The work in the previous two chapters also relies on well-characterized ranging signal and data performance and uses many of the same algorithms as PPP. Constellation monitoring is needed to confirm the assumptions of signal performance that are used in prior chapters as well as ARAIM. Traditional methods do this characterization through the comparison of the broadcast navigation message to precise estimates of the satellite orbit and clock, which are generated by the International GNSS Service (IGS) or National Geospatial-Intelligence Agency (NGA). The results of this approach can be seen in Figure 5.1, which shows the performance history of GPS from 2008-2018, including the five observed fault events. A description of the SU algorithms implementing this approach is described in Appendix A.

However, when these precise estimates are unavailable, these methods fail. If a satellite is observed to be broadcasting a ranging signal with a valid navigation message, it is important that its performance is assessed during that period, regardless of the availability of external precise products. This chapter provides methods for fault detection and nominal performance characterization in the absence of external precise clock estimates. An external precise clock estimate might not be available for
a few reasons. Occasionally, there might be no estimate at all available due to some anomalous behavior that causes the IGS analysis center (AC) to not provide any estimate. However, the most common case, which this chapter is primarily concerned with, is when the IGS AC does not estimate the satellite clock with the signals of interest at all. For example, IGS precise clocks are formed using the L1P-L2P dual frequency pair as a reference, but dual frequency ARAIM users are much more concerned with the performance of the L1 C/A- L5Q dual frequency pair.

The overall goal of such a constellation monitoring system is the long-term characterization of constellation performance. In particular, ARAIM Integrity Support Message (ISM) parameters such as $P_{sat}$ (probability of satellite fault), $P_{const}$ (probability of constellation fault), and bounding User Range Accuracy (URA) sigmas are

![Figure 5.1: GPS Constellation Performance Overview comparing broadcast navigation message to NGA precise orbit and clock: 2008-2018](image)
of interest. Previous papers [84] have described methods of estimating these parameters over long periods and rely heavily on the availability of IGS precise products. However, it has been shown that such precise products are not always available, even during periods where faults have been observed [36] [37]. The goal of IGS analysis centers is accuracy, not necessarily 100% availability, so when anomalous behavior by the satellite is detected, an estimate may not be output. While nominal statistics may not be affected by occasional outages in the precise clock and orbit data, more sensitive parameters such as $P_{sat}$ and $P_{const}$ can be significantly impacted by missing periods of precise data. In addition to data that is missing at times, some products, such as differential code biases (DCB), are only updated at most on a daily basis by the IGS analysis centers [85]. DCB’s, or the corresponding GPS term, the inter-signal corrections (ISC), are required for an ARAIM dual frequency solution. Because of this, the careful characterization of these DCB’s is important. This chapter serves to provide methods to increase the availability of constellation monitoring such that the performance of a satellite can be assessed whenever it is observed to be broadcasting a ranging signal. To support the improvement of constellation monitoring, the work described in this chapter will be provided to the William J. Hughes Technical Center in order to be incorporated into the FAA’s daily GNSS performance evaluation processing.

The approach described in this chapter is to estimate the GNSS clock biases for a dual frequency reference signal as well as estimate DCB’s for additional dual frequency signal pairs of interest. When evaluating constellation performance for ARAIM, the L1 C/A – L5 combination is what we focus on. The benefits of such a process are multiple. First, this allows for the analysis to be decoupled from the IGS clock estimates. Even when IGS estimates are available, this allows for validation of those values. This also allows for analysis that is at a higher rate than is currently available from the IGS analysis centers. For example, GLONASS clock bias estimates are not available at rates higher than 30 seconds. Finally, and perhaps most importantly, the signals combinations of interest can be monitored long term, whereas they have not been properly examined thus far. While GPS DCB’s are generally stable, they have been observed to have changed rapidly when high C/A power is used [76].
current IGS DCB products, the sub-daily impacts on ranging accuracy cannot be assessed.

The clock and DCB estimation process is driven by a ground network of multi-constellation, multi-frequency receivers that are members of the IGS Multi-GNSS Experiment (MGEX) [58] network. Given known positions of the receivers and satellites as well as careful modeling of the range measurements, estimates of the receiver clock biases, satellite clock biases, tropospheric states, carrier phase ambiguities, and DCB’s are produced. The general estimation strategy relies on a low rate batch least squares over long periods and then handing over results from that first estimate to a single epoch estimate of the various clock biases and DCB’s.

5.2 Prior Art

The following works seek to characterize the performance of GNSS broadcast ephemerides, but they rely heavily on precise products generated by IGS or NGA, making them limited in their ability to evaluate L1 C/A and L5 performance.

- Heng [41] introduced scrubbing of the broadcast navigation messages, leading to a much more accurate estimation of the narrow and wide fault rates. Faults are identified both through precise-broadcast comparison as well as by estimating instantaneous URE using receiver observables.

- A multi-constellation study [57] that also examined the new GPS CNAV message explores broad differences between the various GNSS constellations. This study is primarily focused on accuracy, rather than integrity.

- Additional work at Stanford developed methods for determining ARAIM statistics but still rely on NGA and IGS precise products [82] [83].

- Diaz [30] further explores the evolution of the GNSS error statistics over time, but again relies on precise products that only represent a limited set of GNSS signals.
• Zhai [87] sought to produce an independent precise product for ARAIM ISM generation, but uses lower accuracy methods and was left at a covariance analysis.

This next set is more interested in the characterization of GNSS signal qualities, but these studies are not performed with long term characterization in mind, and they are not interested in integrity, for the most part.

• The IGS estimates DCBs daily [55], but these are produced at low rates and are not useful at examining daily effects.

• Daily variations in the clock [56] and code phase [76] have been examined over short periods.

5.3 Clock Bias and DCB Estimation

This section describes the general approach to and results from the Stanford University (SU) clock bias and DCB estimation process.

5.3.1 Ground Receiver Selection

The ground receivers used in the clock and DCB estimation process are selected from the full set of over 200 IGS MGEX receivers. Using the full set of receivers would require more processing power than desired and provide only marginally improved estimation performance over using a subset. In addition, not all receivers even track the signals of interest, making them useless for our purposes. Because of this, we implemented a very simple receiver selection algorithm that allows the clock estimation system to be automated and efficient even as the available receivers changes over time.

The algorithm is a very simple greedy algorithm that first creates a globally distributed grid of “users” at the orbital altitude of the constellation of interest. These users only span from the negative to the positive latitude corresponding to the inclination of the orbits of the constellation of interest. For example, the inclination of the
5.3. CLOCK BIAS AND DCB ESTIMATION

GPS orbits is 55°, so the user grid excludes any users above or below 60° latitudes, where 60° is chosen to have some margin. Ultimately, the station selection is not incredibly sensitive to the amount of margin chosen. From here, given our initially empty station list, the number of users in view from each station is counted, and the station with the most users in view is added to the station list. The number of stations in view from each user is updated with our new station list. The process is repeated, where the additional visibility from each user is computed, except that users with fewer stations in view are weighted more heavily when choosing the new station. This continues until the minimum number of stations in view is reached. In Figure 5.2, the minimum number of stations in view for any user location on orbit is seven, ensuring significant redundancy in the clock and DCB estimates.

![Figure 5.2: Number of ground receivers in view at GPS altitude](image)

5.3.2 Measurement Modeling

Dual frequency code and carrier phase measurements are the primary inputs to the estimator. How exactly the measurements are modeled, i.e. what delays are computed deterministically using models and what are estimated, is very important. The basic pseudorange and carrier phase measurement models are as follows:
CHAPTER 5. CONSTELLATION MONITORING

Dual frequency carrier phase:

\[ \Phi_{if}^{(i,k)} = \|x_s^{(i)} - x_{rx}\| + c(\hat{b}_{rx,c}^{(k)} - \hat{b}_s^{(i,k)}) + m^{(i)}\Delta T - \hat{A}^{(i)} + R_m^{(i)} + \epsilon^{(i)} \]  \hspace{1cm} (5.1)

Dual frequency code phase:

\[ \rho_{if}^{(i,k)} = \|x_s^{(i)} - x_{rx}\| + c(\hat{b}_{rx,c}^{(k)} - \hat{b}_s^{(i,k)}) + m^{(i)}\Delta T + \hat{DCB}_s^{(i,k)} + \hat{DCB}_r^{(k)} + R_m^{(i)} + \epsilon^{(i)} \]  \hspace{1cm} (5.2)

where:

- \( x_s^{(i)} \): satellite position provided by external precise orbit product
- \( x_{rx} \): known receiver position from IGS daily solution
- \( \hat{b}_{rx,c}^{(k)} \): estimated receiver clock bias per DF signal pair
- \( \hat{b}_s^{(i,k)} \): estimated satellite clock bias per DF signal pair and SV
- \( m^{(i)} \): tropospheric mapping function
- \( \Delta T \): estimated delta tropospheric delay
- \( \hat{DCB}_r^{(k)} \): estimated receiver differential code bias per signal (shared across SVs)
- \( \hat{DCB}_s^{(i,k)} \): estimated satellite differential code bias per signal (one per SV and non-reference signal)
- \( R_m^{(i)} \): Other modeled effects. This includes relativistic effects, solid earth tide modeling, satellite antenna phase center offset and variation, ocean loading, modeled tropospheric delay, and carrier phase windup. These are strictly deterministic and not estimated.
- \( \epsilon^{(i)} \): other unaccounted for errors- this should be zero mean and Gaussian, essentially reflecting measurement noise.

The estimated states include the receiver clock biases, satellite clock biases, tropospheric delay delta, differential code biases for each satellite and signal, differential code biases for each receiver and signal, and carrier phase ambiguities for each carrier phase measurement. There only a few differences between this measurement modeling and that of traditional PPP. Namely, that receiver position is not estimated, and that satellite clock biases and DCBs are. For a single receiver, these states would of course
be unobservable, but with a network of receivers, they can be happily estimated.

5.3.3 Evaluating ARAIM Dual Frequency Ranging

Satellite hardware causes varying delays for each signal being broadcast. These delays are typically called the differential code bias; the GPS interface specifications call the terms that describe the delays inter-signal corrections (ISC). One of the goals of this chapter is to evaluate the impact on ranging of these delays on the ARAIM user. The ranging impact is explored in two ways. The first method is to estimate a new clock state using the L1 C/A-L5Q dual frequency code and carrier measurements. This allows for a very similar analysis to what has been done previously, where instead of using IGS precise clock, which use the L1P-L2P dual frequency combination, the SU estimates can be used. In the analysis in this chapter, the difference between the clock bias estimated using L1 C/A and L5Q and the clock bias estimated using L1P-L2P will be primarily examined and compared to broadcast parameters. However, the ARAIM navigation solution uses carrier-smoothed code, so it is very important to examine the ranging impact of using pseudoranges directly. To do this, differential code biases are estimated for the L1 C/A-L5Q combination. Because they are pseudorange-based and thus very noisy, the estimates are averaged over some time span. In this chapter, we use a daily 24-hour interval for validation and finally use a 15-minute interval estimate. The broadcast timing offset to use the L1 C/A – L5 Q combination is found from the following, per IS-GPS-705D [4]:

$$\begin{align*}
PR & = \frac{(PR_{L5Q5} - \gamma_{15}PR_{L1C/A}) + c(ISC_{L5Q5} - \gamma_{15}ISC_{L1C/A})}{1 - \gamma_{15}} - cT_{GD} \quad (5.3) \\
DCB_{L1C/A-L5Q5} & = \frac{c(ISC_{L5Q5} - \gamma_{15}ISC_{L1C/A})}{1 - \gamma_{15}} - cT_{GD} \quad (5.4)
\end{align*}$$

When using any signal combination that is not L1P-L2P, some combination of
ISCs must be used. In this chapter, we compare the measured ranging difference for the L1 C/A-L5Q combination to the broadcast terms that are meant to capture this offset. Ultimately, it is the left hand side term in equation 5.4 that is compared to the clock bias difference or the differential code bias. Figure 5.3 summarizes all of this. The smooth black line at the bottom is the carrier phase-estimated primary clock estimate for the L1P-L2P combination. The red line is a new clock estimated for the L1 C/A –L5Q combination, also estimated using code and carrier. The noisy grey line is the differential code bias. We compare the difference between the red line (clock) or the grey line (DCB) and the black line to the broadcast DCB.

![Figure 5.3: Clock differences and differential code biases](image)

The broadcast DCB terms have been provided by IGS MGEX logs of the CNAV messages. Figure 5.4 shows the DCB terms throughout 2018 for each of the block IIF satellites, which were the only satellites broadcasting on L5. The DCB’s are largely stationary through the year. The jumps in the plot are due to missing broadcast ephemeris data.

### 5.3.4 Estimator Structure

The estimation approach is to use a low rate batch weighted least squares to estimate carrier phase ambiguities and tropospheric delays across the network of ground station network, and then to freeze those estimates and at each epoch, make the final estimates of the various clock and code biases.
The initialization process first requires producing satellite positions for all satellites and times in order to mask measurements from low-elevation satellites. The position of the sun is also precomputed for all times; this is used in the satellite attitude computation for the antenna phase center offset. Antenna phase centers are provided by the IGS [47], and nominal attitude models are used [54]. The effect of these nominal models on the estimation performance when the satellites are in eclipse conditions will need to be investigated in the future. Next, the carrier phase measurements are processed. Cycle slips are detected using a simple test on the change in the geometry-free combination of the two carrier phase measurements that make up the dual frequency observations. Carrier phases are collected into continuous arcs, and only arcs of length exceeding a minimum threshold are kept, and the other measurements are discarded. All of the code phase measurements are kept.

A single batch least squares estimate of all of the desired parameters over all times is not feasible from a memory standpoint, so the estimation is broken into three parts. The first two parts, the “slow” estimator, does estimate all of the desired parameters
but only at a low rate, i.e. 5 or 15 minutes between epochs. The first part, the state initialization, does an extremely rough (200 m accuracy) estimate of the station and satellite clock states using only pseudoranges and no range error modeling. This is done simply to reduce the number of iterations of the precise slow estimator. The states in the precise slow estimator are as follows:

- Receiver clock bias: one per station and epoch
- Satellite clock bias: one per satellite and signal pair and epoch
- Tropospheric delay: one per station and slowly changing over time
- Satellite differential code bias: one per SV, epoch, and non-reference dual-freq. signal combination
- Receiver differential code bias: one per station and non-reference dual-freq. signal combination – this is constant over time.

The reference dual-frequency signal pair used for GPS is the L1P-L2P semi-codeless combination, as is used by both the IGS and the GPS broadcast navigation messages. As such, this combination does not require a satellite differential code bias.

The estimator loops through each station and epoch in order to build the sensitivity matrix for the iterative weighted least squares process. The weights are simply standard σ values for code and carrier phase scaled for the elevation angle. This scaling helps to capture the error due to multipath as well as, more importantly for the carrier phase measurements, the error in the precise orbit estimates, which are projected more heavily onto the line of sight of the measurement at low elevation angles. This estimator does not start with any prior estimates of the receiver or satellite clock biases, which leads to generally needing at least two iterations of the least squares process to converge.

Once the slow estimation is complete, the high rate estimation can occur with the now-frozen carrier phase ambiguities and tropospheric delays. This approach is favorable when compared to doing one enormous iterative least squares because the size of the matrices involved when estimating separately for each epoch do not change,
whereas the size of the matrices involved in a batch estimate increase exponentially. This means that high rates become infeasible for the single batch estimates. For the high rate estimation, a very similar approach is taken to that of the low rate, except that interpolated estimates of the clock biases from the slow estimate are used to seed the process.

5.4 Results

5.4.1 L1P-L2P Clock Estimation Results

![Graph: Daily mean difference in L1P/L2P IF clock bias - SU vs CODE](image)

Figure 5.5: Comparison of SU GPS vs IGS CODE clock bias estimation for 2018

The first set of results is a comparison of the clock output by the SU estimator to that of an IGS analysis center, the Center for Orbit Determination European (CODE). The CODE product is output at 5 second intervals and is centimeter-level accurate. As with all IGS GPS clock products, the reference observables are the L1P and L2P code and carrier, so we compare the CODE clock to our L1P-L2P
clock estimate. Figure 5.6 shows this comparison, where the clock estimate for each satellite is compared to that produce by CODE, and a mean for each day of 2018 is produced for each satellite. Each of the colored dots in Figure 5.6 is a separate satellite and day of 2018. The RMS of the error across all satellites and days is 7 cm, and the daily mean difference never exceeds 35 centimeters over the year. This result is meant simply to be validation of the estimation techniques and data handling so that further results can be trusted.

5.4.2 L1 C/A- L5Q Clock Estimation Results

![Figure 5.6: SU Estimated L1 C/A - L5Q Difference from L1P-L2P](image)

This section examines clock bias difference between the L1P-L2P dual frequency combination and the L1 C/A-L5Q dual frequency combination. Figure 5.6 shows just the difference in clock bias between the two dual frequency pairs as produced by the SU estimator. Each GPS Block IIF satellite is shown as a different set of colored dots. As expected the clock differences are largely static through the year,
but there is a noticeable “swelling” of the clock differences that changes throughout the year that is more prominent on some satellites than others. The values shown in this figure are what will be compared to estimates of the same values produced by the German Aerospace Center (DLR) in Figure 5.7a as well as broadcast in the GPS CNAV message in Figure 5.7b.

Figure 5.7a shows a comparison between the L1 C/A-L5Q clock offset produced by SU and the offset produced by DLR. The daily DCB estimates produced by DLR were subtracted from the values shown in Figure 5.6 to, as before, validate the performance of the estimator. The DCB’s produced by DLR are for each single frequency signal individually, so they must be combined in the proper manner as in Equation 5.4. Additionally, they are daily mean values, so the effects of daily variations are still visible. Despite these factors, the SU estimates closely match the DLR estimates, with an RMS difference over the year of only 10 cm. Figure 5.7b shows a similar comparison that replaces the DLR DCB’s with those produced using CNAV message parameters. Overall, the difference remains relatively small, this time showing an RMS difference of 20 cm. However, the mean differences of some of the satellites, in particular SVNs 66 and 67, push up close to 40 centimeters.

Figure 5.8 shows the same data as in Figure 5.7b, except this time as a histogram for each satellite individually. The individual distributions are tight, with standard
deviations of not more than 10 centimeters for any satellite. The means do approach 40 centimeters in a few cases, as previously mentioned. Ultimately, the effects described in this section are driven by the carrier phase measurements, which are only used for smoothing in ARAIM navigation solutions. Because of this, the impact of the clock difference effects shown here are potentially limited for the ARAIM user.

This section seeks to help explain the daily variations in the clock bias between the L1P-L2P and L1 C/A-L5Q combinations. As they are carrier phase estimates, one might expect that the difference would be extremely stable, but there are noticeable and predictable daily variations. These effects have been observed and described in detail in previous works [56], so this section will be brief. Figure 5.9 takes the data from Figure 5.6 for one satellite and for each day removes the mean value. This leaves only the daily variation of the signal, showing very clearly significant changes in the daily variation throughout the year. The red line in Figure 5.9 is the satellite’s $\beta$ angle, which is the angle between the Sun, the Earth, and the projection of the Sun.
vector onto the satellite orbital plane. The satellite orients itself so as to maintain a solar panel angle with respect to the sun. At high $\beta$ angles, very little yaw motion is required to maintain the specified attitude, whereas at low $\beta$ angles, very rapid noon and midnight turns occur when the satellite passes in front and behind the Earth during the orbit. This yaw motion has been linked to the variations in the clock bias between L5 and other frequencies \cite{56}. More generally, yaw motion during eclipse is generally unpredictable and can lead to errors if the satellite is used in PPP or other non-differential precise applications. Figure 5.10 illustrates the daily variations at high and low beta angles for PRN 10.

5.4.3 DCB Estimation Results

This section shows results relating to the estimation of the differential code bias between the L1 C/A - L5Q and L1P-L2P dual frequency pairs. The previous section examined the difference in the clock bias between the two signal pairs by using the carrier phase, whereas from now on, the code-only difference will be examined. This is particularly important for ARAIM, as it uses a carrier-smoothed code solution,
Figure 5.10: Daily variations in \( \text{L1 C/A - L5Q vs L1P-L2P clock bias} \) at high and low beta angles, and the ranging differences are more significant when considering the true differential code bias.

As before, we seek to validate our results by comparing to known precise estimates of the parameters of interest. In Figure 5.11, daily estimates of the \( \text{L1 C/A-L5Q DCB} \) are compared to the daily estimates of the same values produced by DLR. As before, each set of colored dots indicates a different satellite, and there is one dot per day. The SU estimates are, here, produced at 24 hour intervals and are expected to match those produced by DLR. The RMS difference is 11 cm over the one-year time span. Currently, IGS precise estimates of the DCB’s are only available at 24 hour intervals at the fastest. However, we, and others \cite{76} have observed changes at higher rates, so we have also produced DCB estimates at 15 minute intervals.

Figure 5.12a shows a comparison between the high rate SU DCB estimates and the daily DLR DCB estimates. Immediately, it becomes clear that there is significant deviation from the daily average. This deviation is due to more than noise, though the noise on the estimate does increase. The daily deviation is caused largely by the ongoing high \( \text{C/A} \) campaign, where the broadcast power on \( \text{L1 C/A} \) is increased over certain regions. It is believed that high \( \text{C/A} \) causes the \( \text{L1 C/A} \) chip shape to be
5.4. RESULTS

Figure 5.11: Difference between Daily SU and DLR Differential Code Bias

distorted, leading to ranging biases. Because we are evaluating ranging performance for dual frequency ARAIM, the effect on L1 C/A is amplified when making the ionosphere-free combination. This campaign and its effects on the L1 C/A signal has been described in [76]. We seek to observe the effects over long periods. Figure 5.12b shows the difference between the high rate SU DCB estimates and the CNAV broadcast DCB combination. Some of the error approaches 1 meter on a daily basis due to high C/A. As the accuracy of the broadcast orbit and clock increases and the broadcast URA correspondingly decreases, the effects of the DCB error will become relatively more important.

Figure 5.13 shows the significant variation in the DCB due to high C/A for PRN 6 across a 5 day period in July 2018. PRN 6 exhibits the most extreme changes, which can be seen in the square wave in the DCB. The broadcast CNAV DCB seems to match when high C/A is not used, but when the power is increased, there is a nearly 1 meter ranging difference introduced. This pattern occurs through all of 2018 for many of the IIF satellites.
CHAPTER 5. CONSTELLATION MONITORING

(a) Difference between High Rate SU and DLR DCB

Figure 5.12: Difference between High Rate SU and Other DCB

(b) Difference between High Rate SU and CNAV DCB

The data from Figure 5.12b is put into histograms for each block IIF satellite and shown in Figure 5.14. The effect of high C/A is immediately clear from the multiple bi-modal distributions present. SVN 62, 63, 64, 65, 67, 69, 71, and 72 undergo the high C/A actions, whereas SVN 66, 68, 70, and 73 do not. As a result, the four
satellites that do not execute high C/A have much tighter error distributions when compared to the relatively much broader distributions of those using high C/A. As the capabilities of this tool increase, we will continue to characterize these distributions.

A method for estimating GNSS clock and DCB states has been validated against IGS estimates and found to have sub-decimeter accuracy when comparing using the P1-P2 reference signals. This system has been used to evaluate the ranging performance of the L1 C/A- L5Q dual frequency combination by estimating a separate clock bias and DCB for that signal pair. Daily variations in the clock difference have been observed and linked to the satellite orientation. Daily variations in the DCB have also been observed that have been caused by an ongoing high C/A campaign. The work described in this chapter will be delivered to the FAA’s William J. Hughes Technical Center in order to be incorporated into the daily performance evaluation processing.

Figure 5.14: Histogram of Difference between High Rate SU and CNAV DCB

5.5 Conclusions

A method for estimating GNSS clock and DCB states has been validated against IGS estimates and found to have sub-decimeter accuracy when comparing using the P1-P2 reference signals. This system has been used to evaluate the ranging performance of the L1 C/A- L5Q dual frequency combination by estimating a separate clock bias and DCB for that signal pair. Daily variations in the clock difference have been observed and linked to the satellite orientation. Daily variations in the DCB have also been observed that have been caused by an ongoing high C/A campaign. The work described in this chapter will be delivered to the FAA’s William J. Hughes Technical Center in order to be incorporated into the daily performance evaluation processing.
Chapter 6

Conclusions

6.1 Summary and Conclusions

The underlying theme of this dissertation was the utilization of high precision GNSS techniques for high integrity applications. This is done through the use of PPP filters with multiple hypothesis solution separation as well as through precise estimation of code and clock biases to monitor GNSS signal performance. Today, the use of GNSS in safety critical autonomous systems is limited by the large protection levels and low availability offered by today’s augmentation systems, but the work presented in this thesis can improve both aspects through the following contributions:

6.1.1 PPP with Integrity

The growing field and market of autonomous driving is increasing the demands on GNSS for these safety critical systems. For use in these applications, tight bounds on the positioning error are required. Chapter 3 showed that the fusion of PPP and integrity techniques adapted from those developed for aviation can be used to produce meter-level protection levels for static as well as some automobile environments. Section 3.3 walks through an in depth description of the navigation filter developed to for precision as well as integrity. It consists of banks of PPP filters, where each filter is tolerant to a set of faulted measurements and care is taken to reduce the
computational load of running multiple filters in parallel. The inclusion of an IMU in a tightly coupled fashion was described. An IMU can improve continuity of the protection levels by bridging short gaps in GNSS availability. Section 3.4 demonstrated the system’s capabilities in static, automobile, and flight scenarios. Candidate protection level algorithms were evaluated under static conditions, and an approximate multi-hypothesis solution separation algorithm was ultimately selected for its tight error bounding and computational efficiency. Step and ramp faults were injected into real data in order to test the fault detection and exclusion capabilities. Automobile data in open sky, suburban, and highway environments were evaluated, and protection levels under 2 meters were achieved. In the highway environment in particular, where GNSS continuity breaks are frequent due to the presence of overpasses and large signs, the tightly-coupled IMU filter setup allowed for significantly improved performance over the GNSS-only setup.

6.1.2 Integrity with Broadcast Navigation Data

Many safety critical navigation applications do not have access to the high quality, low latency precise orbit and clock products that are required for the techniques described in Chapter 3. Chapter 4 adapted those techniques for use with the broadcast GNSS navigation message and with or without WAAS clock and ephemeris corrections. Section 4.2.1 described the modifications to the measurement modeling from Chapter 3 that allow for the broadcast navigation message to be used. In particular, states were added that augment the measurement model and track the error in the broadcast ephemeris. The broadcast error state requires a dynamic model. Section 4.3 described the process by which we analyzed the time evolution of the error in the GPS navigation message as experienced by a terrestrial user by comparing the broadcast navigation message outputs to precise clock and ephemeris data. This section also described a method by which jumps in the estimator’s output covariance can be avoided when new navigation messages are provided by multiple satellites simultaneously. A measurement model was also described for the WAAS corrections.

Results were presented for several different scenarios. In Section 4.4.1, protection
levels were compared for algorithms that used precise orbit and clock corrections, the broadcast navigation message, and WAAS corrections. In this section, the navigation filter was initialized with a fault-free position solution, which lead to, after convergence, protection levels of less than five meters in the horizontal directions even with the broadcast ephemeris only. Section 4.4.3 explored the use of these algorithms on hundreds of hours of GPS-Galileo dual-frequency flight data without the fault-free initialization used in the previous section. We compared protection levels produced using the algorithms developed in this chapter to those produced by a snapshot ARAIM prototype. We found that the new sequential algorithms could offer protection levels between 30% and 70% smaller than those from snapshot ARAIM. Finally, a method for fault detection and exclusion was presented that maintains low protection levels even after FDE is executed.

6.1.3 Constellation Monitoring

Chapter 5 introduces techniques by which the signals used by dual-frequency ARAIM users can be monitored and evaluated over long periods. The use of ARAIM requires the careful analysis of the performance of the signal in space, but traditional constellation monitoring techniques rely on precise orbit and clock estimates that were generated using the GPS L1 P(y) and L2 P(y) signals. Because of these, there can be effects that are not captured by these established constellation monitoring techniques. In this chapter, I described, prototyped, and showed results from a estimation and monitoring system that, using a global network of multi-frequency GNSS receivers, is able to estimate precise GNSS clock and differential code biases at high rates. This was used to evaluate the ranging performance of the L1 C/A-L5Q dual-frequency pair through the estimation its own clock bias and high rate differential code bias. The outputs of the full constellation monitoring system, the nominal error bounds and probabilities of fault, can be used in the ARAIM user algorithm as well as the the techniques developed in Chapters 3 and 4.

A year of GPS data was analyzed, where L1P-L2P data was examined for all satellites, and L1 C/A-L5Q data was examined for the block IIF satellites. Section
5.4.1 validated the performance of the estimator by comparing base GPS clock biases estimated to those produced by an IGS analysis center. They agreed to sub-decimeter level, which is adequate for this purpose. Section 5.4.2 showed daily variations in the apparent L1 C/A-L5Q clock bias when compared to the L1P-L2P clock bias. These small, daily differences are potentially due to thermal effects, as the magnitude of the change is correlated with the eclipse season of the satellite. Finally, daily differential code bias differences were examined in Section 5.4.3. A subset of the GPS block IIF satellites were observed to undergo rapid, daily changes in the differential code bias between the L1P-L2P and L1 C/A-L5Q differential code bias. Some of the daily changes approached one meter over the year 2018. These changes were likely caused by a “High C/A” campaign performed by the GPS constellation operators that increases L1 C/A power. The work done in this chapter will be delivered to the FAA’s Hughes Technical Center in order to be incorporated into the daily performance evaluation processing.

6.2 Directions for Future Work

The work done in this thesis sets up a framework and builds out techniques for high accuracy, high integrity navigation, but there is still work to be done to set the parameters involved in the navigation filters. One such parameter is the prior probability of fault. The validity of the protection levels computed is conditional on a conservative threat model. In particular, more work will be needed to assess the appropriate values for the prior probabilities of fault, especially as the vehicle moves through different environments. Similarly, nominal dynamic models for the multipath and broadcast navigation message error states should be further evaluated. The IMU used in the tight coupling is assumed to be fault free, but this is functionally never the case, and some probability of IMU fault should be set and used. In more harsh environments, the probability of fault will likely be higher, which will require the monitoring of additional subsets. As more fault modes are monitored, more computational resources will be required. However, it may be possible to reduce even further the cost of running the sub-filters by exploiting the structure of the problem.
CHAPTER 6. CONCLUSIONS

The work done towards ARAIM constellation monitoring should be incorporated into more general constellation monitoring processes. The code bias distributions found in this thesis are reflected in the ISM parameters generated by the software described in Appendix A. The estimator shown in Chapter 5 is also occasionally subject to erroneous outputs due to limited error checking on the incoming measurements. Further robustness will allow for more confident use of the tool in evaluating probabilities of fault.
Appendix A

Constellation Monitoring ADD

A.1 Algorithm Objective

The purpose of this algorithm is to estimate the instantaneous SISREs in order to support the computation of $R_{sat}$, $R_{const}$, MFD, $P_{sat}$, $P_{const}$, as well as overbounding nominal sigma and bias terms. In order to support these efforts two forms of SISRE are estimated by the algorithm: the Maximum Projected Error (MPE) and the User Projected Error (UPE). MPE is the maximum satellite orbit and clock error projected onto Earth at a particular time. There is one value per healthy satellite at each time epoch. The MPE can only take on the value of zero if the three orbital errors and the clock error are all simultaneously zero. The MPE also can sometimes switch rapidly between positive and negative as the corresponding projections change. As a result, the MPE distribution is bimodal with a notch at zero. Thus the MPE distorts the observed core of the SISRE error distribution. This error distribution is not expected to be Gaussian even if all the underlying clock and ephemeris distributions were Gaussian. The UPE is the projected satellite orbit and clock error at a specific user location. Each user can see many, but not every healthy satellite at any given epoch. Unlike MPE, there will be multiple UPE values per satellite at each epoch (one for each user that has the satellite in view). If the underlying errors are Gaussian, the UPE distributions will also be Gaussian, both at each individual user location and aggregated across all user locations. For this reason, we advocate using UPE as the
error to be used to evaluate the core of the error distributions. Both MPE and UPE are well suited to describe the tail behavior. Figure A.1 shows the MPE determination on the left, 200 evenly distributed user locations around the globe in the middle, and a comparison of the UPE and MPE evaluations on the right. Figure A.2 shows the values of these quantities versus time, the sudden switched in MPE as it changes sign are readily apparent. This density of 200 users has been found to be sufficient such that a value within 2 cm of the MPE will be observed at one or more of the user locations. Other users may see much smaller errors at the same time. Some later analyses will take the maximum value across the two options.

Figure A.1: MPE determination on the left, a grid of users in the middle, and a comparison of MPE and UPE on the right.

Figure A.2: MPE (heavy black line) versus UPEs (multicolored lines) for a particular satellite over time
A.2 Assumptions and Limitations

The ADD assumes that the broadcast navigation and observation data is sufficiently recorded and that common mode recording errors are not present. Therefore, the voting methods will be effective at eliminating outliers. It further assumes that the analysis centers produce accurate orbital estimates, this assumption can be validated by cross comparing ephemeris products, but common mode errors again could be present. This latter effect can be checked against the observation data, but this is dependent on sufficient quantity of good observation data.

Although data is pulled from many sources in order to minimize data gaps, sometimes such gaps cannot be completely eliminated.

A.3 Analyses and Trades

Tradeoffs that may continue to be investigated include:

- Collecting broadcast navigation data from IGS/MGEX versus dedicated network
- Collecting observation data from IGS/MGEX versus dedicated network
- Collecting precise orbit data from NGA/IGS/MGEX versus in house estimate
- Collecting precise clock data from NGA/IGS/MGEX versus in house estimate
- Data collection rate (between 1Hz and once every 15 minutes)

It would be preferable to collect raw navigation data bits from a dedicated global network of receivers. With such a set-up we could record raw data bits regardless of passing parity or other receiver quality checks that may result in discarding data. Voting on raw bits would be less subject to interpretation errors as can be seen in IGS stored navigation data. Raw bits would also provide visibility into such potential issues as broadcast parity failures, non-standard data bits, etc. which are not currently
captured by IGS. Unfortunately we do not currently have access to an adequate global network where we could exercise this level of control.

We currently use external products for precise orbit estimates. We advocate this approach as these products have evolved over years to produce the highest level of accuracy. This performance level is not necessarily easily replicated. Further, these products are widely disseminated and used. Any errors in their content will likely be flagged and updated over time. The disadvantages are that such products could be discontinued or that data gaps may exist that are not deemed worth filling in. Further, there may be inadequate insight into the calculation of these products, and they may contain elements that are not relevant to aircraft processing. For the moment we advocate usage of these products but recommend comparing amongst the different providers. Eventually it may be advantageous to replicate these estimation techniques so that they become more transparent.

Precise clock data is more easily generated given precise orbit estimates and we do advocate this process so that different frequency combinations can be estimated at different rates (presuming that the orbits can be interpolated). This does require access to a large amount of observation data so that outliers may be excluded and the remaining values sufficiently averaged as to produce sufficiently accurate estimates.

Ideally data would be evaluated at 1 Hz (or at a minimum every six seconds). However, this requires a very large volume of data and storage. Initially we worked with 15 minute data and then five minute data. As the process becomes better automated higher rate data should be used.

A.4 Monitor Inputs

The inputs to the SISRE estimation algorithm are described in Table A.1 below.
A.5. MONITOR PROCESSING

A.5.1 External Product Ingestion

A.5.1.1 Precise Orbit Ingestion

Precise orbit data is downloaded from the FTP servers of either the National Geospatial-Intelligence Agency (NGA) or the International GNSS Service (IGS).

<table>
<thead>
<tr>
<th>Symbol/Name</th>
<th>Description</th>
<th>Units</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>year</td>
<td>Year of data to be downloaded</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>doy</td>
<td>Day of year of data to be downloaded</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Server_Flag</td>
<td>Flag indicating data source server (NGA/IGS)</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table A.2: Precise Orbit Ingestion Inputs

Precise orbit and clock data from the NGA can be found at the following FTP site:
ftp.nga.mil/pub2/gps/apcpe/YYYY/apc/

Where YYYY is the year of the desired product. The file format can either be apcWWWWD.exe for a self extracting compressed file or ngaWWWWD.apc for a text file, where WWWW is the GPS week, and D is the day of week.

For multi-GNSS orbit and clock data from MGEX, the following FTP site can be used:


where again, WWWW is the GPS week of the desired data. The format for MGEX orbit data is slightly more complicated:

XXX0MGXFIN_YYYYDDD0000_01D_05M_ORB.SP3.gz

Where YYYY is the year of the desired data, DDD is a three digit day of year padded with zeroes if necessary of the desired data, and XXX is a three character code for the IGS analysis center that produced the data. A good choice here is COD, which represents the Center for Orbit Determination European. Other options can be found at the MGEX website: http://mgex.igs.org/IGS_MGEX_Products.php

For a given day of ephemeris processing, not only should the orbit data from that day be parsed, but the days before and after as well. This prevents issues of large errors being incurred in the interpolation process around the beginning and the end of the day. Precise orbit data is stored in the .sp3 format [Hilla].

A.5.1.2 Precise Clock Ingestion

Precise clock data can be downloaded from IGS FTP servers at higher rates than are available from the precise orbit data files.
A.5. MONITOR PROCESSING

<table>
<thead>
<tr>
<th>Symbol/Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>year</td>
<td>Year of data to be downloaded</td>
<td>-</td>
</tr>
<tr>
<td>doy</td>
<td>Day of year of data to be downloaded</td>
<td>-</td>
</tr>
<tr>
<td>Server_Flag</td>
<td>Flag indicating data source server</td>
<td>-</td>
</tr>
</tbody>
</table>

Table A.3: Precise Clock Ingestion Inputs

Higher rate multi-constellation clock data can be retrieved from the same MGEX FTP directory as the orbit data:
ftp://cddis.gsfc.nasa.gov/pub/gps/products/mgex/

where again, WWWWW is the GPS week of the desired data. The format for MGEX clock data is virtually the same as the orbit data:

XXX0MGXFIN_YYYYDDD0000_01D_30S CLK.CLK.gz

Where YYYY is the year of the desired data, DDD is a three-digit day of year padded with zeroes if necessary, of the desired data, and XXX is a three character code for the IGS analysis center that produced the data.

Clock data is also produced by CODE at a five second rate and is available at the following FTP directory:
cddis.gsfc.nasa.gov/pub/gnss/products/WWWW/

The file format for the high rate data is:
codWWWWD.clk.Z

In this case, D is a one-digit day of week that starts with 0.

Just as for the orbit data, for a given day of ephemeris processing, not only should the clock data from that day be parsed, but the days before and after as well. This prevents issues of large errors being incurred in the interpolation process around the
beginning and the end of the day. Precise clock data is stored in the standard .clk format [67].

A.5.1.3 Receiver Logged Broadcast Navigation Message Ingestion

Navigation messages logged by IGS receivers can be retrieved from the IGS servers.

<table>
<thead>
<tr>
<th>Symbol/Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>year</td>
<td>Year of data to be downloaded</td>
<td>-</td>
</tr>
<tr>
<td>doy</td>
<td>Day of year of data to be downloaded</td>
<td>-</td>
</tr>
<tr>
<td>Server_Flag</td>
<td>Flag indicating data source server</td>
<td>-</td>
</tr>
</tbody>
</table>

Table A.4: Receiver Logged Broadcast Navigation Message Ingestion Inputs

GPS navigation messages as logged by the IGS network receivers can be found at the following FTP directory:

ftp://cddis.gsfc.nasa.gov/gps/data/daily/YYYY/DDD/YYn/

where YYYY is the year of the data, DDD is the zero-padded, three-digit day of year, and YY is the last two digits of the year. The files can either end with the suffix and extension .GN.rnx, indicating that it is a RINEX 3 navigation message file, or it can end with .YYn, indicating that it is a RINEX 2 navigation message file. Either of this are sufficient for GPS processing.

Multi-GNSS navigation message files can be found at:

ftp://cddis.gsfc.nasa.gov/gps/data/daily/YYYY/DDD/YP/

These are all RINEX 3 navigation message files [38].
A.6 Broadcast Navigation Message Voting and Cleansing

The navigation message files logged by the IGS receiver network are subject to frequent logging errors and standard non-compliance. To mitigate this, an algorithm is employed to standardize, cleanse, and vote on the large number of RINEX format navigation message files logged by the individual receivers in the IGS network.

A.6.1 Broadcast Navigation Message Voting Algorithm

Individual navigation message files can be subject to errors in the rounding of parameters, non-standard storage of various terms (i.e. logging the URA index rather than URA upper bound value), duplicated messages, and other errors that can significantly impact the evaluation of such sensitive terms as $R_{sat}$. The voting and cleansing process starts by parsing hundreds of RINEX broadcast navigation files. The cleansing algorithm then recovers the least significant bit of each broadcast navigation term and converts stored URA values that appear to have logged the URA index rather than the upper bound. Duplicated messages are removed for each receiver. Finally, given the relatively clean set of messages compiled across all receivers, the true broadcast navigation message for each Issue of Data is voted on.

<table>
<thead>
<tr>
<th>Symbol/Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRDC List</td>
<td>Set of RINEX broadcast navigation files</td>
<td>-</td>
</tr>
</tbody>
</table>

Table A.5: Navigation Message Voting Inputs

Further detail regarding these algorithms can be found in [41].

<table>
<thead>
<tr>
<th>Symbol/Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>suglXXXX.YYn</td>
<td>Cleaned RINEX broadcast navigation file</td>
<td>-</td>
</tr>
</tbody>
</table>

Table A.6: Navigation Message Voting Outputs
A.7 Broadcast Navigation Message Evaluation

The objective of this section is to perform the direct computation of the difference between the precise orbit and clock estimates and the broadcast orbit and clock estimates at the epochs of interest. The precise clocks must be synchronized to the broadcast clocks, and the broadcast navigation message error will be computed in the ECEF as well as local reference frames. Finally, a worst case URE estimate will be generated.

A.7.1 Precise Orbit Interpolation

In order to perform a direct comparison between the precise satellite orbits and the broadcast satellite orbits, both must be aligned in time. The precise orbits can be interpolated using a polynomial interpolation scheme in order to match the rate of the clock products.

<table>
<thead>
<tr>
<th>Symbol/Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRNs</td>
<td>N-length vector of PRN per desired interpolation</td>
<td>-</td>
</tr>
<tr>
<td>epochs</td>
<td>N-length GPS epoch per desired interpolation</td>
<td>sec</td>
</tr>
<tr>
<td>Ppos</td>
<td>Precise position matrix from IGS or NGA products</td>
<td>m</td>
</tr>
<tr>
<td>Pepochs</td>
<td>GPS epochs associated with Ppos</td>
<td>sec</td>
</tr>
<tr>
<td>nPolyFit</td>
<td>Number of points to use for the polynomial fit</td>
<td>-</td>
</tr>
<tr>
<td>pFit</td>
<td>Order of fit for polynomial interpolation</td>
<td>-</td>
</tr>
</tbody>
</table>

Table A.7: Precise Orbit Interpolation Inputs

A polynomial fit can be used to interpolate the precise orbit from the low rate at which it is provided to a higher rate to match the clock products. An 8th order polynomial fit is used with the 12 orbital positions that are nearest in time to the desired interpolation time. This process is described in more depth in [32]. If velocity estimates were not provided in the precise orbit data, interpolated velocity can be found by simply perturbing the interpolation time by some amount and taking a simple first difference in the output positions.
A.7. BROADCAST NAVIGATION MESSAGE EVALUATION

<table>
<thead>
<tr>
<th>Symbol/Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>posP</td>
<td>Interpolated satellite precise position</td>
<td>m</td>
</tr>
<tr>
<td>velP</td>
<td>Interpolated satellite precise position</td>
<td>m/s</td>
</tr>
</tbody>
</table>

Table A.8: Precise Orbit Interpolation Outputs

A.7.2 Broadcast Navigation Message Propagation

For each epoch and satellite, the most recently broadcast navigation message is selected, and the contained information is propagated to the epoch. The propagation is done per the ICD provided by the CSP.

<table>
<thead>
<tr>
<th>Symbol/Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>prn</td>
<td>N-length vector of PRN per desired propagation epoch</td>
<td>-</td>
</tr>
<tr>
<td>GPSweek</td>
<td>N-length GPS week per desired propagation epoch</td>
<td>week</td>
</tr>
<tr>
<td>GPSsec</td>
<td>N-length GPS time of week per desired propagation epoch</td>
<td>sec</td>
</tr>
<tr>
<td>eph</td>
<td>Structure containing navigation message clock and ephemeris parameters</td>
<td>Mixed</td>
</tr>
</tbody>
</table>

Table A.9: Broadcast Navigation Message Propagation Inputs
### Table A.10: Broadcast Navigation Message Propagation Outputs

<table>
<thead>
<tr>
<th>Symbol/Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>ECEF x position</td>
<td>m</td>
</tr>
<tr>
<td>y</td>
<td>ECEF y position</td>
<td>m</td>
</tr>
<tr>
<td>z</td>
<td>ECEF z position</td>
<td>m</td>
</tr>
<tr>
<td>x_dot</td>
<td>ECEF x velocity</td>
<td>m</td>
</tr>
<tr>
<td>y_dot</td>
<td>ECEF y velocity</td>
<td>m</td>
</tr>
<tr>
<td>z_dot</td>
<td>ECEF z velocity</td>
<td>m</td>
</tr>
<tr>
<td>clock_bias</td>
<td>Satellite clock bias</td>
<td>s</td>
</tr>
<tr>
<td>clock_drift</td>
<td>Satellite clock drift</td>
<td>s/s</td>
</tr>
<tr>
<td>accuracy</td>
<td>Broadcast URA value</td>
<td>m</td>
</tr>
<tr>
<td>health</td>
<td>Broadcast health flag</td>
<td>-</td>
</tr>
<tr>
<td>IODC</td>
<td>Broadcast Issue of Data Clock</td>
<td>-</td>
</tr>
<tr>
<td>AoD</td>
<td>Age of Data- time since the Time of Ephemeris</td>
<td>s</td>
</tr>
<tr>
<td>tslu</td>
<td>Time since last upload</td>
<td>s</td>
</tr>
<tr>
<td>toe_m_ttom</td>
<td>Time of Ephemeris minus Time Tag of Message</td>
<td>s</td>
</tr>
<tr>
<td>Fit_interval</td>
<td>Broadcast navigation message fit interval</td>
<td>s</td>
</tr>
</tbody>
</table>

#### A.7.3 Clock Synchronization

Precise clocks produced by the NGA are typically very tightly synchronized with GPS time, but this is not necessarily the case for clock data produced by the IGS. A bulk clock offset does not affect position accuracy, so it is removed in this step. Steps are taken to mitigate the impact of outliers.
A.7. BROADCAST NAVIGATION MESSAGE EVALUATION

<table>
<thead>
<tr>
<th>Symbol/Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>clkP</td>
<td>Precise clock estimates for each satellite and epoch</td>
<td>sec</td>
</tr>
<tr>
<td>clock_bias</td>
<td>Broadcast clock estimates for each satellite and epoch</td>
<td>sec</td>
</tr>
</tbody>
</table>

Table A.11: Clock Synchronization Inputs

One can impose the constraint that the GPS clock biases estimated by the IGS analysis center are, as a whole, aligned with the broadcast clock biases. Put another way, the mean error between the IGS clocks and the broadcast clocks is zero. This constraint can be imposed because any bulk offset in timing would be reflected as a receiver clock bias to the user, which does not impact positioning. If there are anomalous clock bias estimates, removing a mean that includes those clocks could result in multiple satellites appearing to be anomalous rather than just the one. In order to mitigate this, those clocks can be screened by checking the difference of each clock error from the median clock error. If that difference exceeds a preset threshold, it shall be removed from the bulk timing offset computation. [41] also utilizes a more complex iterative least squares approach.

<table>
<thead>
<tr>
<th>Symbol/Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>clkP_aligned</td>
<td>Synchronized precise clock estimates for each satellite and epoch</td>
<td>sec</td>
</tr>
</tbody>
</table>

Table A.12: Clock Synchronization Outputs

A.7.4 Satellite Orbital Attitude Adjustment with Antenna Phase Center Offset

Satellite orbital estimates produced by the NGA refer to the antenna phase center of the satellite, while the position estimates produced by the IGS analysis centers refer to the center of mass of the satellite. The broadcast navigation message refers to the antenna phase center. If IGS data is used, the CoM data must be adjusted to refer to
the antenna phase center. This is done using nominal satellite attitude models, where
the satellite yaws in order to maintain a pointing angle with respect to the sun.

<table>
<thead>
<tr>
<th>Symbol/Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \vec{x}_{CoM} )</td>
<td>Precise satellite center of mass position in ECEF</td>
<td>m</td>
</tr>
<tr>
<td>( \vec{x}_{CoM \rightarrow APC} )</td>
<td>Center of mass to antenna phase center offset in satellite body frame</td>
<td>m</td>
</tr>
<tr>
<td>( \vec{x}_{sun} )</td>
<td>Sun position in ECEF</td>
<td>m</td>
</tr>
</tbody>
</table>

Table A.13: Satellite Orbital Attitude Adjustment with Antenna Phase Center Offset

Inputs

GNSS satellites nominally maintain a yaw-steering attitude where the broadcast antenna is nadir pointing and the satellite yaws to point the solar panels towards the sun. This yaw-steering frame is generally aligned with the body frame, except during eclipse season. For more information about satellite behavior during eclipse season, see [54]. The yaw-steering frame can be defined as follows:

\[
\hat{e}_{x,YS} = \hat{e}_{y,YS} \times \hat{e}_{z,YS} \\
\hat{e}_{y,YS} = \frac{\vec{e}_{sun} \times \vec{x}_{sat}}{\| \vec{e}_{sun} \times \vec{x}_{sat} \|} \\
\hat{e}_{z,YS} = -\frac{\vec{x}_{sat}}{\| \vec{x}_{sat} \|}
\]

The rotation matrix between the body and ECEF frame can then be constructed.

\[
R_{ECEF}^{body} = [\hat{e}_{x,YS} \hat{e}_{y,YS} \hat{e}_{z,YS}]
\]

The antenna phase center position in ECEF is the center of mass position offset by the position of the antenna phase center relative to the center of mass expressed in the ECEF frame.

\[
\vec{x}_{APC} = \vec{x}_{CoM} + R_{body}^{ECEF} \vec{x}_{CoM \rightarrow APC}
\]
A.7. **BROADCAST NAVIGATION MESSAGE EVALUATION**

<table>
<thead>
<tr>
<th>Symbol/Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \vec{x}_{APC} )</td>
<td>Precise satellite antenna phase center position in ECEF</td>
<td>m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Symbol/Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \vec{x}_p )</td>
<td>Precise satellite position in ECEF</td>
<td>m</td>
</tr>
<tr>
<td>( \vec{v}_p )</td>
<td>Precise satellite velocity in ECEF</td>
<td>m</td>
</tr>
</tbody>
</table>

Table A.14: Satellite Orbital Attitude Adjustment with Antenna Phase Center Offset Outputs

### A.7.5 ECEF Orbit and Clock Difference Computation

Once the satellite positions and clocks are aligned, the difference between the two can be computed. This is a simple subtraction between the two.

### A.7.6 Local Reference Frame Conversion

The difference between the broadcast and precise orbits is typically related to the physics of the satellite orbits, and rotating the position error to the local radial-long track- cross track (RAC) frame can provide more insight. The RAC frame can be found using the satellite position and velocity vectors.

The RAC frame can be computed given the ECEF position and velocity of the satellite. The frame consists of unit vectors in the radial direction, the cross product between the radial vector and the velocity vector, which produces the cross-track vector, and the along-track vector completes the right-handed triad.

\[
\hat{r} = \frac{-\vec{x}_{sat}}{||\vec{x}_{sat}||}
\]

\[
\hat{c} = \frac{\hat{r} \times \vec{v}_{sat}}{||\hat{r} \times \vec{v}_{sat}||}
\]
\[ \hat{a} = \hat{c} \times \hat{r} \]

The rotation matrix can be constructed from the individual unit vectors:

\[ R_{ECEF}^{RAC} = [\hat{r} \ \hat{a} \ \hat{c}] \]

<table>
<thead>
<tr>
<th>Symbol/Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_{ECEF}^{RAC} )</td>
<td>Rotation matrix from the ECEF frame to the RAC frame</td>
<td>-</td>
</tr>
</tbody>
</table>

Table A.16: Local Frame Conversion Outputs

### A.7.7 Maximum Projected Error Computation

An analytical worst-case user error for a terrestrial user can be computed given a specific elevation mask.

<table>
<thead>
<tr>
<th>Symbol/Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \vec{x}_p )</td>
<td>Precise satellite position in ECEF</td>
<td>m</td>
</tr>
<tr>
<td>( \vec{x}_b )</td>
<td>Broadcast satellite position in ECEF</td>
<td>m</td>
</tr>
<tr>
<td>( \delta \vec{x} )</td>
<td>Precise clock bias minus broadcast clock bias</td>
<td>s</td>
</tr>
</tbody>
</table>

Table A.17: Geometric method for computing Maximum Projected Error
A.8. ERROR AGGREGATION

A.8.1 Classification of Faulted and Non-faulted Epochs

The comparison between precise and broadcast orbit and clock at each epoch must first be classified. The classifications of non-faulted data is as follows:

1. exist_segments: a mapping exists from PRN to SVN at this time for this SVN

2. valid_segments: an unfaulted, successful comparison has occurred at this time

<table>
<thead>
<tr>
<th>Symbol/Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPE</td>
<td>Worst case ranging error for a terrestrial user</td>
<td>m</td>
</tr>
</tbody>
</table>

Table A.18: MPE Outputs
3. **noeph_notruth_segments**: a mapping exists from PRN to SVN, but no truth and no broadcast ephemeris are available

4. **unhealthy_segments**: the satellite has been determined to be set unhealthy

5. **eph_notruth_segments**: broadcast ephemeris is available but no precise product

6. **truth_noeph_segments**: precise product is available but no broadcast

There are several criteria that must be met for data to be classified as faulted. The satellite must be set healthy, the time must be within the navigation message fit interval, and the maximum projected error must exceed the fault threshold determined by the URA/SISA value.

### A.8.2 Nominal Statistics Generation

#### A.8.2.1 Projection onto User Grid (UPE)

The ECEF satellite position error and clock error are projected onto the line of sight of 200 evenly spaced global terrestrial users. Projections that lie below a certain local user elevation threshold are removed.

<table>
<thead>
<tr>
<th>Symbol/Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>truePos</td>
<td>Matrix of precise satellite positions over time interval</td>
<td>m</td>
</tr>
<tr>
<td>brdcPos</td>
<td>Matrix of broadcast satellite positions over time interval</td>
<td>m</td>
</tr>
<tr>
<td>trueClk</td>
<td>Matrix of precise satellite clock biases over time interval</td>
<td>s</td>
</tr>
<tr>
<td>brdcClk</td>
<td>Matrix of broadcast satellite clock biases over time interval</td>
<td>s</td>
</tr>
<tr>
<td>$\vec{x}_{user}$</td>
<td>Matrix of user locations to project error onto</td>
<td>m</td>
</tr>
</tbody>
</table>

Table A.19: Nominal Statistics Generation Inputs
The first step to computing the UPE for each user, \( j \), and satellite, \( i \), is to compute the line of sight vector.

\[
\hat{r}_{\text{los}} = \frac{\hat{\vec{x}}^{(i)}_{\text{sat,true}} - \hat{\vec{x}}^{(j)}_{\text{user}}}{\|\hat{\vec{x}}^{(i)}_{\text{sat,true}} - \hat{\vec{x}}^{(j)}_{\text{user}}\|}
\]

Project the orbit and clock error onto the line of sight vector:

\[
UPE_{\text{los}}^{(i)(j)} = \hat{r}_{\text{los}} \frac{\hat{\vec{x}}^{(i)}_{\text{sat,true}} - \hat{\vec{x}}^{(i)}_{\text{sat,brdc}}}{\|\hat{\vec{x}}^{(i)}_{\text{sat,true}} - \hat{\vec{x}}^{(j)}_{\text{sat,brdc}}\|}
\]

Compute the elevation angle for this line of sight. If it is lower than a specified mask angle, it should be removed. Once the line of sight broadcast navigation message error is computed for each satellite, they can be stored for histogram generation. Another option at this point is to remove the mean error across all satellites for a given user and time. This can be done because the common mode error would be absorbed into the receiver clock bias, but it does lead to a difference between the maximum UPE and the MPE. Figure 3.1.52 can be compared to Figure 3.1.12 to observe the difference in maximum UPE computed with and without the receiver clock offset removed.
Figure A.4: MPE (heavy black line) versus UPEs (multicolored lines) for a particular satellite over time with mean error removed.

<table>
<thead>
<tr>
<th>Symbol/Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>usrProjErr</td>
<td>Matrix of satellite orbit and clock errors projected onto user line of sights over time interval</td>
<td>m</td>
</tr>
</tbody>
</table>

Table A.20: Nominal Statistics Generation Outputs

A.8.2.2 Error Partitioning

The error, once projected onto the user LOS’s, can be partitioned in various ways before being placed into a histogram format. Several different partitions are made as described in the “outputs” table.
A.8. ERROR AGGREGATION

<table>
<thead>
<tr>
<th>Symbol/Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpe</td>
<td>Maximum projected user error matrix</td>
<td>m</td>
</tr>
<tr>
<td>usrProjErr</td>
<td>Matrix of satellite orbit and clock errors projected onto user line of sights over time interval</td>
<td>m</td>
</tr>
<tr>
<td>racErr</td>
<td>Satellite orbital error in RAC frame</td>
<td>m</td>
</tr>
<tr>
<td>clkErr</td>
<td>Satellite clock error matrix</td>
<td>s</td>
</tr>
</tbody>
</table>

Table A.21: Error Partitioning Inputs

A number of different histograms are collected, each partitioned in different manners and/or containing different data. The first is called $mpe_{pdf\_data}$, which is an MPE histogram partitioned by satellite, URA, and time since last upload (TSLU), where TSLU is binned by two hour windows. When the word “partitioned” is used, it is meant that the data is split into multiple categories, and these partitions are then made into different dimensions of the histogram matrix. For example, $mpe_{pdf\_data}$, being partitioned by satellite, URA, and TSLU, is a four-dimensional matrix, where the first dimension partitions across the various satellites, the second across the URAs, the third across the TSLUs, and the fourth dimension is the error bins. Another histogram is $rng_{pdf\_data}$, which is the URA-normalized instantaneous error from each satellite projected onto 200 evenly spaced global users. This is also partitioned by satellite, URA, and TSLU.

The error computed for each user is the previously described UPE. $rng_{pdf\_daily}$ also contains the normalized instantaneous user projected error, but it is partitioned by satellite and day. This is particularly useful for compiling statistics by day or month. $hist_{chi\_data}$ is the sum of squared errors partitioned by user and number of satellites in view (degrees of freedom). This only considers nominal data- major faults are removed. Finally, simple histograms of the radial, long-track, cross-track, and clock bias are generated, only partitioned by satellite.

Initialize histogram matrix (nSatellites x nURAs x nTSLU x nBins) to zeros

for tdx = 1:nSteps
for udx = 1:nUsers

    Build user-satellite line of sight in ECEF
    Build user-satellite LOS in local ENU frame
    Project broadcast orbit and clock error onto user line of sight
    Remove LOS’s below elevation mask
    Collect error into various histograms

end
end

<table>
<thead>
<tr>
<th>Symbol/Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpe_pdf_data</td>
<td>MPE histogram partitioned by satellite, URA, and TSLU binned into two hour windows</td>
<td>m</td>
</tr>
<tr>
<td>rng_pdf_data</td>
<td>URA-normalized user projected error histogram partitioned by satellite, URA, and TSLU</td>
<td>m</td>
</tr>
<tr>
<td>rng_pdf_daily</td>
<td>URA-normalized user projected error histogram partitioned by satellite and day</td>
<td>m</td>
</tr>
<tr>
<td>hist_ch_data</td>
<td>Sum of squared error histogram partitioned by user and number of satellites in view</td>
<td>m</td>
</tr>
<tr>
<td>rac_hist_data</td>
<td>Satellite orbit error histogram in radial, along track, and cross track directions</td>
<td>m</td>
</tr>
</tbody>
</table>

Table A.22: Error Partitioning Outputs

A.9 Monitor Outputs

The outputs from the SISRE estimation algorithm are described in Table A.23 below.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Units</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Times when satellite errors exceed threshold</td>
<td>PRNs and time intervals when either MPEs or UPEs exceed 4.42 URA</td>
<td>Seconds since Jan 6, 1980</td>
<td>$R_{\text{sat}}$, $R_{\text{const}}$, MFD determination.</td>
</tr>
<tr>
<td>MPE Histograms</td>
<td>Histograms of MPE data</td>
<td>various</td>
<td>URA Bounding</td>
</tr>
<tr>
<td>UPE Histograms</td>
<td>Histograms of MPE data</td>
<td>various</td>
<td>URA Bounding</td>
</tr>
<tr>
<td>Chi-square Histograms</td>
<td>Histograms of chi-square data</td>
<td>various</td>
<td>URA Bounding</td>
</tr>
</tbody>
</table>

Table A.23: Monitor Outputs
Appendix B

Odometry and PPP with Integrity

B.1 Introduction

This study investigates the benefits and challenges of the use of vehicle wheel odometry in a solution separation system that already includes PPP and IMU. Section B.2 introduces the models for the wheel odometry measurements as well as the vehicle motion constraints. Section B.3 introduces the experimental dataset provided by Hexagon and explores the position error growth and protection level impacts from using different subsets of the available sensor suite.

B.2 Methods

The PPP algorithm with solution separation has been developed using banks of extended Kalman filters that use dual frequency code phase, carrier phase, and doppler measurements. Additional details regarding the specific implementation can be found in [34] [35], but the implementation is largely a standard tightly coupled PPP-IMU filter.
This study introduces additional sensors and pseudomeasurements into the solution separation PPP engine. Figure B.1 shows the sensor suite utilized in the experiments in Section B.3. In addition to GNSS and IMU measurements, wheel ticks from the rear wheels and constraints on the vehicle motion are placed on the cross-track and vertical velocities of the car.

The odometry measurements come in the form of wheel tick counts integrated over time. Every second, the tick count increases by an integer amount. This can be modeled as the velocity at the wheel integrated over time and scaled by some amount. Here, it was found that the wheel diameter is approximately 31.5 inches with 1000 ticks per revolution or some variation of those two parameter that would produce the same base scale factor. The ticks are scaled up by the base scale factor in order to get to units of meters, and they are used directly in the extended kalman filter.

![Figure B.1: Vehicle Sensors](image)
The measurement innovation for the right wheel can be produced:

$$
\delta z_{\text{rear-right}} = N_{\text{ticks}} * s_0 (1 - s_{\text{rear-right}}) - \dot{v}_{\text{rear-right}}
$$  \hspace{2cm} (B.1)

Where $N_{\text{ticks}}$ is the number of ticks counted by the odometer since the last update, and $s_0 = 30.5 \times 0.0254 \times \pi / 1000$, and $\dot{v}_{\text{rear-right}}$ is the integrated velocity at the location of the rear right wheel, and $s_{\text{rear-right}}$ is an estimated scale factor parameter. The velocity integration is performed during the IMU mechanization step, which occurs at the rate of the IMU measurements. In this case, it is 125 Hz. The velocity integration is as follows:

$$
\dot{v}_{\text{rear-right}} = \frac{1}{\tau_0} \int_{t-\tau_0}^{t} \left[ -1 \ 0 \ 0 \right] \left[ \dot{C}_e^b(t') \dot{v}_e^b(t') + (\omega_b^e \times l_{b,\text{rear-right}}^b) \right] dt'
$$  \hspace{2cm} (B.2)

Where $\tau_0$ is the measurement update time step (here, 1 second), $\dot{C}_e^b$ is the estimated rotation matrix from the ECEF to body frame, $\dot{v}_e^b$ is the estimated instantaneous velocity at the IMU reference location, $\omega_b^e$ is the angular rate in the body frame, and $l_{b,\text{rear-right}}^b$ is the lever arm offset from the IMU reference point to the rear right wheel.

The measurement sensitivities with respect to various states can be produced. The sensitivity for the odometry measurement w.r.t. the velocity, attitude, and scale factor are as follows:

$$
H_{\text{rear-right},v} = -\frac{1}{\tau_0} \int_{t-\tau_0}^{t} \left[ -1 \ 0 \ 0 \right] \dot{C}_e^b(t') \dot{v}_e^b(t') \times dt' \hspace{2cm} (B.3)
$$

$$
H_{\text{rear-right},\Psi} = -\frac{1}{\tau_0} \int_{t-\tau_0}^{t} \left[ -1 \ 0 \ 0 \right] \dot{C}_e^b(t') dt' \hspace{2cm} (B.4)
$$
The innovation and sensitivities for the rear left wheel can be computed in the same manner.

The non-holonomic constraints are computed in much the same way as the odometry measurements. Velocity in the cross-track and vertical directions are integrated in the IMU mechanization step at high rate, except in this case, the pseudo-measurement innovations are simply the integrated velocities in each of the directions. The cross-track integrated velocity and the sensitivities can be computed as follows:

\[
\hat{v}_{cross,\text{rear-right}} = \int_{t-\tau_0}^{t} \left[ \begin{array}{ccc} 0 & 1 & 0 \\ \end{array} \right] \left[ \begin{array}{c} \dot{C}_e^b(t') \dot{v}_e^b(t') + (\omega_{\text{eb}} \times l_{b,\text{rear-right}}) \end{array} \right] dt' 
\]

\[
H_{cross,v} = -\frac{1}{\tau_0} \int_{t-\tau_0}^{t} \left[ \begin{array}{ccc} 0 & 1 & 0 \\ \end{array} \right] \left[ \begin{array}{c} \dot{C}_e^b(t') \dot{v}_e^b(t') \end{array} \right] \times dt' 
\]

\[
H_{cross,\Psi} = -\frac{1}{\tau_0} \int_{t-\tau_0}^{t} \left[ \begin{array}{ccc} 0 & 1 & 0 \\ \end{array} \right] \dot{C}_e^b(t') dt' 
\]

Similarly, the vertical integrated velocity and the sensitivities can be computed as follows:

\[
\hat{v}_{vertical,\text{rear-right}} = \int_{t-\tau_0}^{t} \left[ \begin{array}{ccc} 0 & 0 & 1 \\ \end{array} \right] \left[ \begin{array}{c} \dot{C}_e^b(t') \dot{v}_e^b(t') + (\omega_{\text{eb}} \times l_{b,\text{rear-right}}) \end{array} \right] dt' 
\]

\[
H_{vertical,v} = -\frac{1}{\tau_0} \int_{t-\tau_0}^{t} \left[ \begin{array}{ccc} 0 & 0 & 1 \\ \end{array} \right] \dot{C}_e^b(t') \dot{v}_e^b(t') \times dt' 
\]
\begin{equation}
H_{\text{vertical,}\psi} = -\frac{1}{\tau_0} \int_{t-\tau_0}^{t} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \dot{C}_e^{b}(t') dt'
\end{equation}

Finally, the state vector used in this error state extended Kalman filter is:

State vector:

\[
\begin{bmatrix}
\hat{x}_{\text{IMU}} \\
\hat{v}_{\text{IMU}} \\
\hat{\Phi} \\
\hat{a}_{\text{bias}} \\
\hat{\omega}_{\text{bias}} \\
\hat{b}_{rx,c} \\
\hat{d}_{rx,c} \\
\Delta T \\
\hat{s}_{\text{rear-right}} \\
\hat{s}_{\text{rear-left}} \\
D\hat{C}_{B_{rx}}^{(j)} \\
F\hat{D}\hat{C}_{B_{rx}}^{(i,j)} \\
\hat{A}^{(i,j)} \\
\hat{I}^{(k)}
\end{bmatrix}
\]
where: $\hat{x}_{IMU}$: Position of IMU in ECEF
$\hat{v}_{IMU}$: Velocity of IMU in ECEF
$\hat{\Psi}$: Attitude Euler angles
$\hat{a}_{bias}$: Accelerometer bias
$\hat{\omega}_{bias}$: Gyro bias
$\hat{b}_{rx,c}$: Receiver clock bias
$\hat{d}_{rx,c}$: Receiver clock drift
$\Delta T$: Tropospheric delta
$\hat{s}_{rear-right}$: Odometry scale factor for rear right wheel
$\hat{s}_{rear-left}$: Odometry scale factor for rear right wheel
$\hat{FDCB}^{(i,j)}_{rx}$: Receiver differential code bias term for each GLONASS SV
$\hat{A}^{(i,j)}$: Carrier phase ambiguities for each SV
$\hat{f}^{(k)}$: Ionospheric delay for single frequency signals

B.3 Results

B.3.1 Dataset Description

Figure B.2: Dataset Location Satellite View
The dataset used in the experiments described in this report was recorded from an automobile driving in a loop for approximately 35 minutes. Figure B.3 shows the ground track of the path according to the truth data. The vehicle drives in a loop that extends approximately 60 meters in the North-South direction and only 25 meters in the East-West direction. There is very little vertical change throughout. The satellite image shown in figure B.2 indicates that this was recorded in a vacant lot of some sort.

![Ground Track of Vehicle Dataset](image)

Figure B.3: Ground Track of Vehicle Dataset

Importantly, there is a significant outage in the GNSS measurements during the data recording. The outage starts approximately 7 minutes into the dataset, and it lasts for approximately 20 minutes. The GNSS outage, shown in figure B.4, impacts all but two of the satellites in view at the onset of the outage and all of the satellites at
end of the outage. It does not seem to impact the ranging errors of the measurements significantly as the outage begins or ends. During the GNSS outage, both IMU and odometry measurements are still available.

![Figure B.4: Gap in GNSS Observation Availability](image)

The dataset comes from a receiver, IMU, and odometry setup mounted on an automobile traveling in a lot in Calgary, Alberta, Canada. The receiver is a NovAtel OEM 7500 using GPS (L1 C/A – L2P semi-codeless) and GLONASS (L1 C/A- L2P) at 1 Hz. Code phase, carrier phase, and Doppler measurements are used. The IMU is a tactical grade IMU operating at 125 Hz. The lever arm offset between the GNSS antenna phase center and the IMU has been surveyed. The displacements between the IMU reference point and the center of both the front and rear axles have also been surveyed in addition to the wheel base and wheel track lengths. Wheel odometry data in the form of raw tick counts for each of the rear wheels is used at 1 Hz.
**B.3.2 PPP Only**

Figure B.5: Position Error and Protection Level for estimator with GNSS Only

For the first set of results, the estimator is configured to only use GNSS measurements. Because of the lengthy GNSS outage, no protection level is produced for most of the 35 minutes of overall data. However, when GNSS is available, decimeter accuracy is achieved by the all in view filter. After the first 7 minutes of convergence, protection levels of less than 1.3 meters are produced in the East and North directions, and a vertical protection level of 2.4 meters is produced. A few features can be observed in Figure B.5. The typical PPP convergence after initialization leads to a rapid decrease in the protection levels. At the end of the first GNSS window, there is a slight uptick in protection level, which is caused by the loss of most, but not all pseudoranges for a single second. The position error rapidly grows after measurements become unavailable as the simple dynamics model leads to the estimated position leaving in a straight line. Protection levels are not produced during this period, and after
ten seconds without measurements, the entire filter simply resets and waits for new measurements in order to reinitialize. Importantly, the position error never exceeds the protection levels.

Figure B.6 shows the position error of the AIV solution in the heavy black line as well as the position error of each of the monitored subsets in the colored lines. The GLONASS code phases seem to exhibit some biases, so a DCB is estimated for each satellite. The AIV solution is biased by more than 2 meters compared to truth at the initialization, which is performed with the code phase measurements. The red line at the bottom of the bottom plot in Figure B.6 is the subset with GLONASS PRN 2 excluded. The geometry of that satellite seems to make it relatively influential, and without estimating the DCB or otherwise accounting for it, the biased solution that exists at the initialization can persist and even lead to a violation of the protection levels. Ultimately, it is important to properly model the measurements in order to achieve the desired protection levels.

Figure B.6: All in View and Subset Position Errors- GNSS Only
B.3.3 PPP with IMU and Constraints

The introduction of IMU and motion constraints significantly reduces position error as well as protection levels, shown in Figure B.7. The motion constraints, which keep the vehicle from slipping in the cross-track direction and keep the body-vertical velocity zero, are implemented as unfaulted measurements- i.e. they are assumed to never fault and thus do not have subsets monitoring faults on the measurements. An experiment with more data could investigate the actual slip rate and introduce a fault mode that would also be monitored. Not including the constraints was also briefly investigated, but as expected, there was significant position error (> 1 km) accumulation over the interval, and ultimately seemed less than useful.

Figure B.7: Position Error and Protection Level for estimator with PPP + IMU + Constraints

The constraints are modeled as the integrated velocity in the cross track and vertical directions, as described in the previous section. In order to calibrate the
B.3. RESULTS

measurement noise, they were initially deweighted significantly, and during the peri-
ods with GNSS available, the residuals were examined, and a measurement noise of
5 cm integrated over each second was found to conservatively bound the residuals.
This is what is used in the full experiment setup as the measurement noise. Import-
tantly, it was also found that the vertical velocity constraint residuals were biased
slightly, which would lead to even more significant vertical drift during the GNSS
outage. It was found that this could be corrected through a very slight (< 1 degree)
pitch correction between the IMU frame and the vehicle body frame. Without this
correction, the vertical protection levels would be exceeded. As with the GLONASS
DCBs described in the previous section, such mismodeling could either be estimated
or calibrated offline. In this case, to reduce the implementation complexity and be-
cause of the limited amount of data, the correction was simply computed offline and
added to the model.

The protection levels have been significantly reduced, where the protection levels
in the East and Up directions are the most significantly impacted. This seems to
be the result of the constraints. The constraints reduce the error in the cross track
and the vertical directions. Of course, the vehicle stays essentially horizontal through
the data collection, so the vertical velocity constraint suppresses error growth to an
extent throughout. The vehicle is mostly traveling in the North-South direction, so
the cross-track constraint suppresses error growth in the East-West direction, leading
to decreased error and protection levels.
B.3.4 PPP with IMU, Constraints, and Odometry

The inclusion of rear wheel odometry serves to further reduce protection levels. The integration of odometry measurements looks very similar to the velocity constraints used in the previous section in that the provided measurements are the integrated velocity in the forward direction at the location of the wheels in the body frame. In particular, odometer measurements constrain the error growth in the forward direction, which is visible in the reduction of the protection levels and error in the North direction in Figure B.8. The calibration procedure for the measurement noise for the odometry was the same as for the slip constraints, where the measurements were deweighted and residuals examined. Using a modeled wheel diameter of 31.5 inches leads to scale factors that remain approximately 0, but the dynamics of the scale factor states could be tweaked in the future.

Figures B.9, B.10, and B.11 explore the impact of a significantly shorter GNSS
B.3. RESULTS

initialization time. Nominally, there is more than 6 minutes of time with GNSS measurements, and the various scale factors and IMU biases are estimated during this period. These three figures show the impact of starting the estimator at a much later point, where only 10 seconds of GNSS are available before the outage. Because of this, those scale factors are not able to properly converge before the outage begins, and this leads to much more significant error growth during the outage. Figure B.9 shows that the error actually exceeds the protection level towards the end of the outage. It seems that the covariance propagation is not conservative enough, and there may be additional uncertainty needed in the uncertainties related to the IMU and odometer at the initialization.

Figure B.9: Position Error and Protection Level for estimator with PPP + IMU + Constraints + Odometry with only 10 seconds GNSS initialization

Figure B.11 shows the resulting position error on the ground track, which appears to be a counterclockwise rotation of the traveled loop. Figure B.11 illustrates estimated biases that are not resolved by the time the GNSS outage begins. These plots
show the position error of the AIV filter in black and the position error of the subsets in the colored lines. Notably, at the bottom of the bottom plot again there is a subset that is significantly deviated from the other subsets, and this is again a GLONASS satellite where the per-satellite DCB state has not been resolved in the short period with GNSS available. Such unconverged terms without the appropriate uncertainties leads to error growth during the GNSS outage.

Figure B.10: Ground track for estimator with PPP + IMU + Constraints + Odometry with only 10 seconds GNSS initialization

<table>
<thead>
<tr>
<th>Case</th>
<th>Min PL [m]</th>
<th>Max PL after conv. [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPP</td>
<td>E 1.38</td>
<td>N 1.13</td>
</tr>
<tr>
<td></td>
<td>U 2.38</td>
<td>E INF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>N INF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>U INF</td>
</tr>
<tr>
<td>PPP+IMU</td>
<td>E 1.37</td>
<td>N 1.08</td>
</tr>
<tr>
<td></td>
<td>U 2.32</td>
<td>E 18.92</td>
</tr>
<tr>
<td></td>
<td></td>
<td>N 64.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>U 10.06</td>
</tr>
<tr>
<td>PPP+IMU+ODO</td>
<td>E 1.38</td>
<td>N 1.09</td>
</tr>
<tr>
<td></td>
<td>U 2.31</td>
<td>E 12.74</td>
</tr>
<tr>
<td></td>
<td></td>
<td>N 14.85</td>
</tr>
<tr>
<td></td>
<td></td>
<td>U 10.02</td>
</tr>
</tbody>
</table>

Table B.1: Odometry Protection Level Summary
Figure B.11: All in View and Subset Position Errors for estimator with PPP + IMU + Constraints + Odometry with only 10 seconds GNSS initialization

B.4 Conclusions

Odometry section conclusions: Vehicle sensors were integrated into a PPP-IMU solution separation system, and the impacts on the protection levels produced were explored. The results are summarized in Table B.1. Some important points that arose from the experiments are that there a number of biases and calibrations that needed to be accounted for in order to produce valid protection levels, including differential code biases, IMU mounting angles, IMU biases, and odometry scale factors. Protection levels were limited to less than 15 meters in all directions over an extended GNSS measurement outage through the use of all available sensors and vehicle constraints.


[58] Oliver Montenbruck, Peter Steigenberger, Lars Prange, Zhiguo Deng, Qile Zhao, Felix Perosanz, Ignacio Romero, Carey Noll, Andrea Stürze, Georg Weber, and et al. The multi-gnss experiment (mgex) of the international gnss service


[76] Peter Steigenberger, Steffen Thölert, and Oliver Montenbruck. Flex power on gps block iir-m and iif. GPS Solutions, 23(1), Aug 2018.


