Design and evaluation of integrity algorithms for PPP in kinematic applications

Kazuma Gunning, Juan Blanch, Todd Walter, *Stanford University*

Lance de Groot, Laura Norman, *Hexagon Positioning Intelligence*

ABSTRACT

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. Up to now, these techniques have mostly been used to provide high accuracy, rather than focusing on high integrity applications. In this paper we apply the methodology and algorithms used in aviation to determine position error bounds with high integrity (or protection levels) for a PPP position solution.

0. INTRODUCTION

Precise Point Positioning techniques [1] 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 [2] 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 RTK has dealt with the ambiguity resolution process under nominal error conditions [3], [4], and less on the integrity of the position solution under fault conditions.

We start (Section 1) with an overview of our PPP filter implementation, and a description of the threat model (Section 2). In Section 3, we describe two classes of integrity algorithms: solution separation and sum of squared residuals based (also called residual based (RB) –which is a misnomer, as all autonomous integrity monitors are based on the residuals.) Section 4 presents the data sets that were used to evaluate the algorithms. In Section 5, we compare the PLs obtained with different algorithms. In section 6, we present the results obtained with the most promising PL formulation in four different data sets: static, dynamic in open sky conditions, dynamic in midtown suburban conditions, and in flight.

1. PPP ALGORITHM OVERVIEW

PPP techniques achieve small position solution errors by using precise clock and ephemeris corrections and applying models of the remaining errors (both temporal and spatial). Our sequential PPP filter implementation is based on a simple extended Kalman filter (EKF) with estimated parameters comprising the receiver position, clock biases for each constellation in use, a tropospheric delay, float ambiguities for each tracked carrier phase, and multipath contributions. Dual-frequency measurements can be incorporated from GPS, GLONASS, Galileo, and BeiDou.

The precise orbit and clock estimates are drawn from the Center for Orbit Determination in Europe (CODE) which is an IGS MGEX analysis center. We also implemented a batch least squares PPP that, while not useful for real-time applications, can be used for static and kinematic truth position generation. The PPP filters developed have been shown to perform with centimeter-level accuracy for static scenarios and decimeter-level accuracy for kinematic scenarios.

As we will see later, the implementation of solution separation requires a bank of parallel filters. Running completely separate EKF's for each subset is too computationally demanding. The code was structured such that most modeled effects are only computed once based on the all-in-view solution. These model outputs are shared across all subsets, thus reducing computational load significantly:

- Several effects (solid tides, relativity) are not sensitive to position and clock solution
- Range components based on state estimates (geometric range, receiver clock bias, carrier phase ambiguities, tropospheric wet delay) are computed on a per-subset basis

The time updates and measurement updates are still computed separately per subset. Figure 1 illustrates the process.

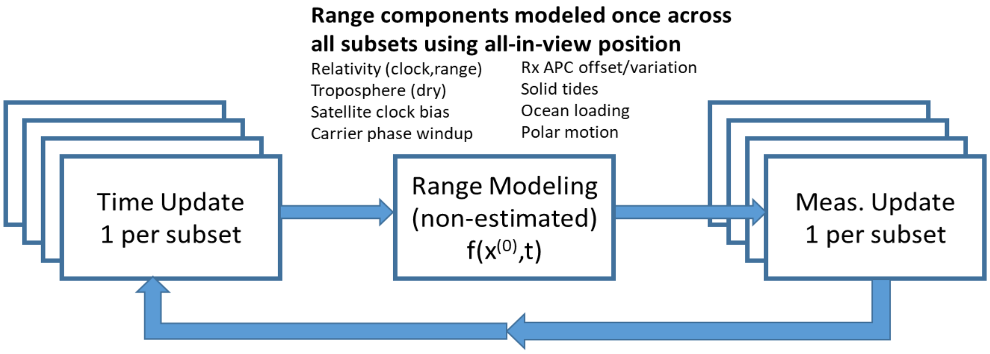


Figure 1. Bank of filters in PPP implementation

2. NOMINAL ERRORS AND THREAT MODELS

In order to quantify the integrity risk of the position solution and its error bound, it is essential to have:

- a trusted model of the nominal errors
- a model of the potential threats.

The nominal model describes the expected measurements when no faults are present. The threat model is a list of the fault modes with their probabilities. The fault modes are all the conditions that that can cause the errors not to follow the nominal model.

We note that we can decide whether a fault mode is accounted for in the nominal error model or listed in the threat model. For example, we can either inflate the sigmas to account for very large multipath, or list it as an additional threat.

For this paper, we will assume that faults will occur at a rate of 10^{-4} per hour and will persist for at most an hour. These fault modes account for clock and ephemeris errors, errors in the correction terms, very large multipath, and undetected cycle slips. This is a preliminary threat model that will need to be refined.

The measurements used in the EKF are single and dual frequency code and carrier phase measurements along with single frequency Doppler measurements. Continuity of L1 carrier phase measurements is significantly better than that of L1/L2 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 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:

$$\Phi_{if}^{(i)} = \left\|x_s^{(i)} - \hat{x}_{rx}\right\| + c\left(\hat{b}_{rx,c} - b_s^{(i)}\right) + m^{(i)}\widehat{\Delta T}^{(i)} + b_{pwu}^{(i)} - \hat{A}^{(i)} + \mathrm{R_m} + \widehat{M}^{(i)} + \hat{\epsilon}^{(i)} \tag{1}$$

Dual frequency code phase:

$$\rho_{if}^{(i)} = \left\|x_s^{(i)} - \hat{x}_{rx}\right\| + c\left(\hat{b}_{rx,c} - b_s^{(i)}\right) + m^{(i)}\widehat{\Delta T}^{(i)} - \widehat{DCB}_{rx}^{(i)} - f_j\widehat{FDCB}_{rx}^{(i)} + \mathrm{R_m} + \widehat{M}\ ^{(i)} + \hat{\epsilon}^{(i)} \tag{2}$$

Single frequency carrier phase:

$$\Phi\ ^{(i)} = \left\|x_s^{(i)} - \hat{x}_{rx}\right\| + c\left(\hat{b}_{rx,c} - b_s^{(i)}\right) + m^{(i)}\ \widehat{\Delta T}^{(i)} + b_{pwu}^{(i)} - \hat{A}^{(i)} - \mathrm{I^{(i)}} + \mathrm{R_m} + \widehat{M}^{(i)} + \hat{\epsilon}^{(i)} \tag{3}$$

Single frequency code phase:

$$\rho\ ^{(i)} = \left\|x_s^{(i)} - \hat{x}_{rx}\right\| + c\left(\hat{b}_{rx,c} - b_s^{(i)}\right) + m^{(i)}\ \widehat{\Delta T}^{(i)} - \widehat{DCB}_{rx}^{(i)} - f_j\widehat{FDCB}_{rx}^{(i)} + \mathrm{I^{(i)}} + \mathrm{R_m} + \widehat{M}^{(\mathrm{i})} + \hat{\epsilon}^{(i)} \tag{4}$$

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 orbit product

$m^{(i)}$- tropospheric mapping function

$\widehat{\Delta T}^{(i)}$- estimated delta tropospheric delay

$b_{pwu}^{(i)}$- carrier phase wind-up

$\hat{A}^{(i)}$- estimated float carrier phase ambiguity

$\widehat{M}^{(i)}$- estimated multipath delay on the signal

$\widehat{DCB}_{rx}^{(i)}$- estimated receiver differential code bias per signal (shared across SVs)

f_j- GLONASS signal frequency channel number from -7 to 6

$\widehat{FDCB}_{rx}^{(i)}$- estimated frequency-dependent GLONASS differential code bias per signal (shared across SVs)

$\mathrm{I^{(i)}}$- ionospheric delay/advance

$\mathrm{R_m}$- 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

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 reflecting multipath error on code and carrier phase measurements, which was modeled as first order Gauss Markov process

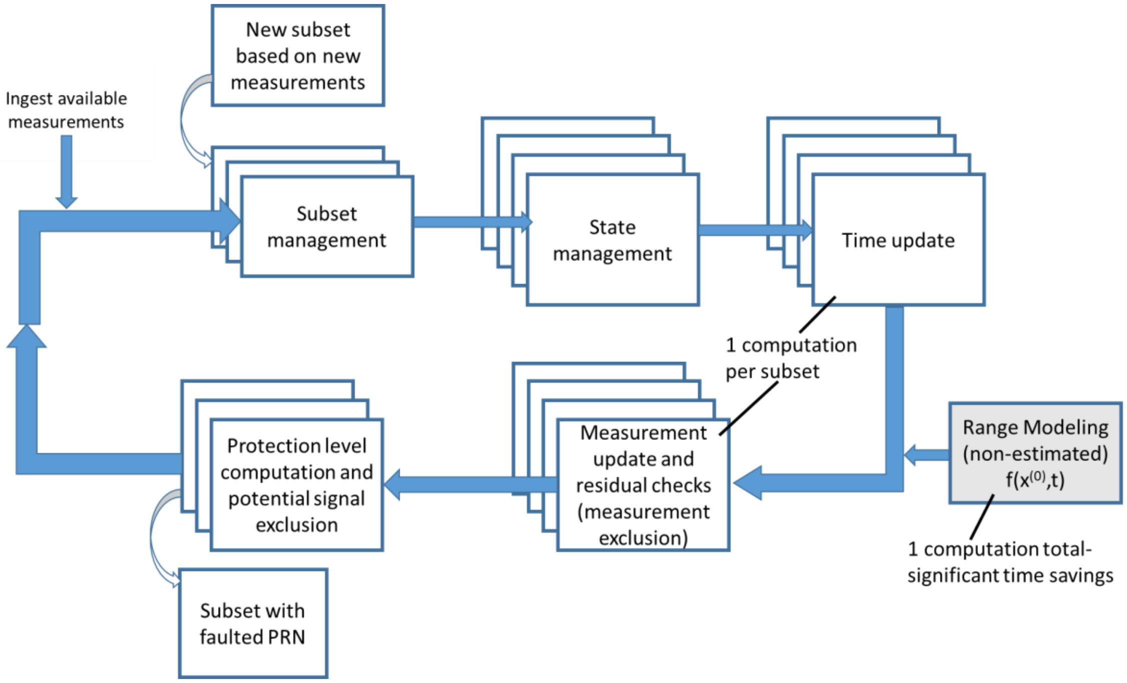
$$M^{(i)}(t+\tau) = M^{(i)}(t)e^{-\frac{\tau}{T_{mp}}} \tag{5}$$

The time constant (T_{mp}) was set to 100 seconds per previous multipath analysis [5].

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, 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 $0.01^2\,m^2$ multiplied by the time step.

$$\hat{E}^{(i)}_{carrier} = \widehat{M}^{(i)}_{carrier} + \hat{A}^{(i)} + \hat{\epsilon}^{(i)}_{carrier} \tag{6}$$

2.1 Implementation Description



New subset based on new measurements
Ingest available measurements
Subset management
State management
Time update
1 computation per subset
Range Modeling (non-estimated) $f(x^{(0)},t)$
1 computation total- significant time savings
Measurement update and residual checks (measurement exclusion)
Protection level computation and potential signal exclusion
Subset with faulted PRN

Figure 2. PPP with Solution Separation Implementation Diagram

The implementation of the PPP algorithm with solution separation uses an extended Kalman filter. The EKF loop starts with measurement ingestion. If any measurements are from satellites that have not been seen by the EKF before, then new subsets and states must be introduced. The new states consist of error and carrier phase ambiguity states.

Parameter	**Initial Sigma (meters)**
Clock bias	100
Position	10
Velocity	10
Tropospheric Wet Delay	0.1
Float carrier phase ambiguity	100

Code phase multipath	2
Carrier phase error state	0.05
Differential code bias	20
Frequency-Dependent DCB	1

Table 2.1. Initial state uncertainties

The time update, wherein the state and covariance matrices are propagated from the previous time step to the current, is very straightforward. The position, velocity, and acceleration states are simply propagated forward together. All other states are assumed to be static except for the code multipath state, which is modeled as a first order Gauss-Markov process as previously mentioned. The time update is performed individually for each subset state and covariance, but it is a relatively computationally cheap operation, as it is only matrix multiplication. The process noise values used in the static and automotive scenarios are listed in Table 2.2.

Parameter	**Process Noise (m)**
Clock bias	100
Forward Position	0.77
Upward Position	0.11
Rightward Position	1.18
Velocity	1
Acceleration	0.5
Tropospheric Wet Delay	$0.002^2/3600^{1/2}$
Carrier phase error state	0.01
Code phase multipath	0.2
Differential Code Bias	0
Frequency-Dependent DCB	0

Table 2.2. Process noise settings

In preparation for the measurement update step, many computations must be done to produce predicted pseudorange and carrier phase measurements. These computations consist of, for example, the precise satellite orbit position and clock offset, solid earth tide effects, and tropospheric delays. The other included models are described in Section 2. In order to reduce computation time, these computations are done only once per EKF iteration, and the outputs are shared across all subsets. These effects are not sensitive to the small variations in receiver position across the various subsets, so this introduces very little error, but it reduces the computational load significantly.

Measurement	**Sigma (m)**
Pseudorange	2
Carrier phase	0.03
Doppler	0.05

Table 2.3. Measurement noise settings

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 final 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. 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 need not be included in the fault rate and are instead caught at the measurement update.

In addition to GNSS measurements, pseudo-measurements are occasionally introduced as constraints on the motion. When measurements are temporarily unavailable, as occurs occasionally when, for example, driving under a bridge, an altitude measurement based on the last available altitude estimate is introduced to constrain the position estimate over short periods. Doing so keeps the covariance from increasing as much as it otherwise would. This added measurement is based on the assumption that, over a period of 1-2 seconds, the altitude of an automobile will not change significantly. The variance of the altitude measurement is 10^2 cm^2 summed with the position covariance in the “up” direction. The actual measurement is the current altitude estimate. These measurements are only used for short gaps of a few seconds, after which, they are no longer included.

3. INTEGRITY MONITORING ALGORITHMS AND PROTECTION LEVELS

The protection level (PL) is determined by:

- the threat model
- the nominal model
- the test statistic
- the integrity risk allocation
- the upper bounds used to simplify the computation

The PL is defined such that:

$$\text{Prob}\left(\text{position error} > PL \,\&\, \text{test passes}\right) \leq \text{Integrity risk allocation} \tag{7}$$

3.1 Integrity Risk

Let us develop Equation (7) using the formula of total probability:

$$\text{Prob}\left(\left|e^T\left(x-\hat{x}\right)\right| > PL \,\&\, y \in \Omega\right) = \sum_{i=0}^{N} \text{Prob}\left(\left|e^T\left(x-\hat{x}\right)\right| > PL, y \in \Omega \,|\, H_i\right) P\left(H_i\right) \tag{8}$$

Where

x is the state

$\hat{x}$ is all-in-view estimate of the state

y is the vector of measurements. For Kalman filter solutions, it refers to all measurements up to time t, as well as the propagation equations

Ω is the region (in the space of measurements) defined by the detection statistics

e is the vector that extracts the coordinate of interest from the state vector

H_i is the fault hypothesis *i*

The probability P(H_i) is the prior probability of the fault mode. As indicated above, in this paper we will assume a value of 10^{-5}.

3. 2 Kalman filter solutions and snapshots solutions

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 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 [6]:

$$\sigma_{ss}^{(i)2} = \sigma^{(i)2} - \sigma^{(0)2} \tag{9}$$

Where

$\sigma^{(i)2}$ is 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. It also allows us to update the worst case slope in the case of residual based RAIM.

3.3 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.

3.3.1 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:

$$\Omega_{ss} = \left\{ y \,|\, \left| e^T \left(\hat{x} - \hat{x}^{(i)} \right) \right| \leq T_i \right\} \tag{10}$$

where $\hat{x}^{(i)}$ is 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:

$$T_i = Q^{-1}\left(\alpha_i P_{fa} \right) \sigma_{ss}^{(i)} \tag{11}$$

Where

$\sigma_{ss}^{(i)}$ is the standard deviation of the solution separation under nominal conditions, and

α_i is the allocation of the probability of false alert to the fault mode. The sum over all modes must not exceed one.

Q is the complement of the normal cdf, and Q^{-1} its inverse.

3.3.2 Sum of squared residuals statistic

The detection region for the sum of the square residuals is defined by:

$$\Omega_{\chi^2} = \left\{ y \,|\, y^T \left(W - WG\left(G^T W G \right)^{-1} G^T W \right) y \leq T_{\chi^2} \right\} \tag{12}$$

Where:

G is the batch observation matrix (and therefore including the propagation step)

W is the inverse of the measurement covariance

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

$$
\begin{aligned}
P\left(y \notin \Omega_{\chi^2}\right) &= P\left\{y^T\left(W - WG\left(G^TWG\right)^{-1}G^TW\right)y > T_{\chi^2}\right\} \\
&= 1 - F_{\chi^2}\left(T_{\chi^2}, n-p\right) = P_{FA}
\end{aligned}
\tag{13}
$$

where $F_{\chi^2}\left(K, dof\right)$ is the cdf of a chi-square distribution with dof degrees of freedom

n is 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.

3.3.3 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:

$$\chi^2(t) = \chi^2(t-1) + \left(y_t - G_t\hat{x}_{t|t-1}\right)^T\left(W_t - W_tG_tP_{t|t}G_t^TW_t\right)\left(y_t - G_t\hat{x}_{t|t-1}\right) \tag{14}$$

Or:

$$\chi^2(t) = \chi^2(t-1) + \left(y_t - G_t\hat{x}_{t|t}\right)^T\left(W_t + W_tG_tP_{t|t-1}G_t^TW_t\right)\left(y_t - G_t\hat{x}_{t|t}\right) \tag{15}$$

where we have used the following notations:

$\chi^2(t)$ is 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 (14) and (15) is in Appendix A.

3.4 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 [7], 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 [6] and [8])
- Second upper bound is a simplification of the previous one

3.4.1 Integrity risk for one fault mode for solution separation

It can be shown that a tight upper bound of the integrity risk for one fault mode is given by [6]:

$$P\left(\left|e^T\left(x-\hat{x}\right)\right| \geq PL, y \in \Omega_{ss} \mid H_i\right) \leq \max_{\beta} Q\left(\beta - \frac{T_i}{\sigma_{ss}^{(i)}}\right)\left(Q\left(\frac{PL-\beta\sigma_{ss}^{(i)}}{\sigma^{(0)}}\right)\right) + Q\left(\frac{PL}{\sigma^{(0)}}\right) \tag{16}$$

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

$$F_{SS}\left(\frac{PL-T_i}{\sqrt{\sigma^{(0)2}+\sigma_{ss}^{(i)2}}}, \frac{\sigma_{ss}^{(i)}}{\sigma^{(0)}}\right) = \max_{\beta} Q\left(\beta - \frac{T_i}{\sigma_{ss}^{(i)}}\right)\left(Q\left(\frac{PL-\beta\sigma_{ss}^{(i)}}{\sigma^{(0)}}\right)\right) \tag{17}$$

where F_{ss} is a function of two variables determined numerically [6].

3.4.2 Exact PL for solution separation

Going back to Equation (8) and using Equations (16) and (17), we obtain the following implicit definition of the PL:

$$2Q\left(\frac{PL_{SS,exact}}{\sigma^{(0)}}\right) + \sum_{i=0}^{N} F_{SS}\left(\frac{PL_{SS,exact}-T_i}{\sigma^{(i)}}, \frac{\sigma_{ss}^{(i)}}{\sigma^{(0)}}\right) P\left(H_i\right) = PHMI \tag{18}$$

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 [6], [8].

3.4.3 First upper bound for solution separation

This upper bound is the one that was chosen to analyze Advanced RAIM performance in [8], and is the default proposed in [6]. We have the upper bound:

$$F_{SS}\left(\frac{PL-T_i}{\sigma^{(i)}}, \frac{\sigma_{ss}^{(i)}}{\sigma^{(0)}}\right) \leq Q\left(\frac{PL-T_i}{\sigma^{(i)}}\right) \tag{19}$$

We now replace it in (18), and obtain:

$$2Q\left(\frac{PL_{SS,approx1}}{\sigma^{(0)}}\right)+\sum_{i=0}^{N}Q\left(\frac{PL_{SS,approx1}-T_i}{\sigma^{(i)}}\right)P\left(H_i\right)=PHMI \tag{20}$$

This equation is also solved used a half interval search but is much faster than (18), because we use the complement of the normal cdf *Q* instead of *F_{SS}*.

3.4.4 Second upper bound for solution separation

This upper bound is defined by:

$$PL_{SS,approx2}=\max\left(T_i+Q^{-1}\left(\frac{PHMI}{N*P\left(H_i\right)}\right)\sigma^{(i)}\right) \tag{21}$$

A complete proof of the integrity of this PL can be found in [6]. It is based on the inequality:

$$Q\left(\frac{PL_i-T_i}{\sigma^{(i)}}\right)\leq\frac{PHMI}{N*P\left(H_i\right)} \tag{22}$$

3.4.5 Choice of thresholds

For a given P_{FA}, any choice of the allocations that meets $\sum_i \alpha_i=1$ (see Equation (11)) is valid. The default chosen in [6],[8] 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 (20)) is possible ([7]), it is computationally expensive. We use instead a suboptimal scheme which consists in making all the thresholds equal:

$$T=Q^{-1}\left(\alpha_i P_{fa}\right)\sigma_{ss}^{(i)}$$

$$\sum_i \alpha_i=1 \tag{23}$$

3.5 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.

3.5.1 Slope formula

The slope of a given mode is linked to the standard deviation of the solution separation statistic [7],[9] through the equation:

$$slope_i=\frac{\sigma_{ss}^{(i)}}{\sigma^{(0)}} \tag{24}$$

This means that it can be computed recursively using the subset filter covariance update via Equation (9).

3.5.2 *Integrity risk for one fault mode for the sum of squared residuals*

For the sum of squared residuals, the integrity risk for one fault mode can be computed using the formula ([10]):

$$P\left(\left|x-\hat{x}\right|\ge PL, y\in\Omega_{\chi^2} \mid H_i\right) = \max_{\lambda} P_{ncx}\left(T_{\chi^2}, n-p, \lambda^2\right)\left(Q\left(\frac{PL-\lambda\sigma_{ss}^{(i)}}{\sigma^{(0)}}\right)+Q\left(\frac{PL+\lambda\sigma_{ss}^{(i)}}{\sigma^{(0)}}\right)\right) \tag{25}$$

Where

$P_{ncx}\left(K, dof, \lambda^2\right)$ is the non-central chi-square cdf with non-centrality parameter λ^2 and *dof* degrees of freedom.

We note:

$$F_{\chi^2}\left(\frac{PL}{\sigma^{(0)}}, \frac{\sigma_{ss}^{(i)}}{\sigma^{(0)}}, T_{\chi^2}, n-p\right) = \max_{\lambda} P_{ncx}\left(T_{\chi^2}, n-p, \lambda^2\right)\left(Q\left(\frac{PL-\lambda\sigma_{ss}^{(i)}}{\sigma^{(0)}}\right)+Q\left(\frac{PL+\lambda\sigma_{ss}^{(i)}}{\sigma^{(0)}}\right)\right) \tag{26}$$

3.5.3 Exact protection level for sum of squared residuals

The "exact" protection level is computed using the integrity risk equation (8) and (26):

$$2Q\left(\frac{PL_{chi2,exact}}{\sigma^{(0)}}\right)+\sum_{i=0}^{N} F_{\chi^2}\left(\frac{PL_{chi2,exact}}{\sigma^{(0)}}, \frac{\sigma_{ss}^{(i)}}{\sigma^{(0)}}, T_{\chi^2}, n-p\right)P\left(H_i\right) = PHMI \tag{27}$$

As for the exact solution separation *PL*, we use a half interval search.

3.5.4 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 [6]. It is defined by:

$$2Q\left(\frac{PL_{chi2,approx1}}{\sigma^{(0)}}\right)+\sum_{i=0}^{N} Q\left(\frac{PL_{chi2,approx1}-\sigma_{ss}^{(i)}\sqrt{T_{\chi^2}}}{\sqrt{\sigma^{(0)2}+\sigma_{ss}^{(i)2}}}\right)P\left(H_i\right) = PHMI \tag{28}$$

3.5.5 Second upper bound for the sum of squared residuals statistic

This coarser upper bound uses the same approach as Equation (21):

$$PL_{chi2,approx2} = \max\left(\sigma_{ss}^{(i)}\sqrt{T_{\chi^2}}+\sqrt{\sigma^{(0)2}+\sigma_{ss}^{(i)2}}\,Q^{-1}\left(\frac{PHMI}{N*P\left(H_i\right)}\right)\right) \tag{29}$$

3.5.6 Robustness against assumptions

It is important to note that the PLs defined by (20), (21), (28), and (29) 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 [11],[12], or [13]) of the nominal errors.

This is not the case for the PLs defined by (19) and (27) which rely on exact gaussian distributions, in particular on the fact that the test statistic is independent of the error estimate.

4. DATASETS

We used four data sets, one corresponding to a static receiver in open sky conditions, one corresponding to open sky road conditions, one corresponding to a suburban road conditions, and one corresponding to flight conditions.

4.1 Static data set

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

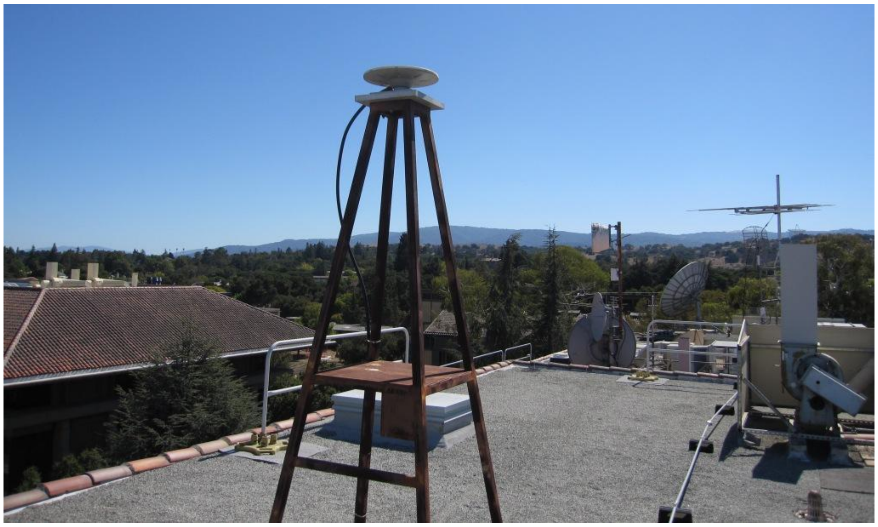


Figure 4.1 Receiver antenna used to collect the static data at Stanford University

4.2 Dynamic data set in open sky conditions

The first dynamic data 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.

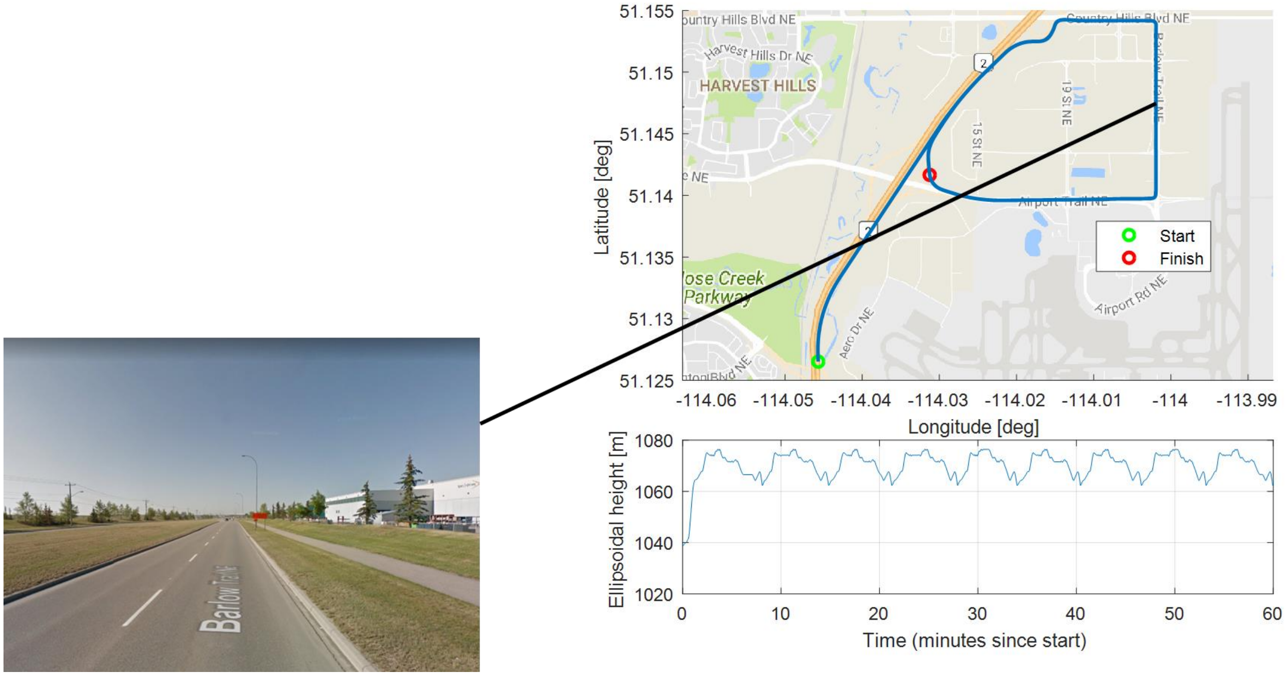


Figure 4.2 Dynamic data set collected in Calgary (open sky conditions)

- Receiver: NovAtel OEM 7500
- 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

4.3 *Dynamic data set in midtown suburban environment*

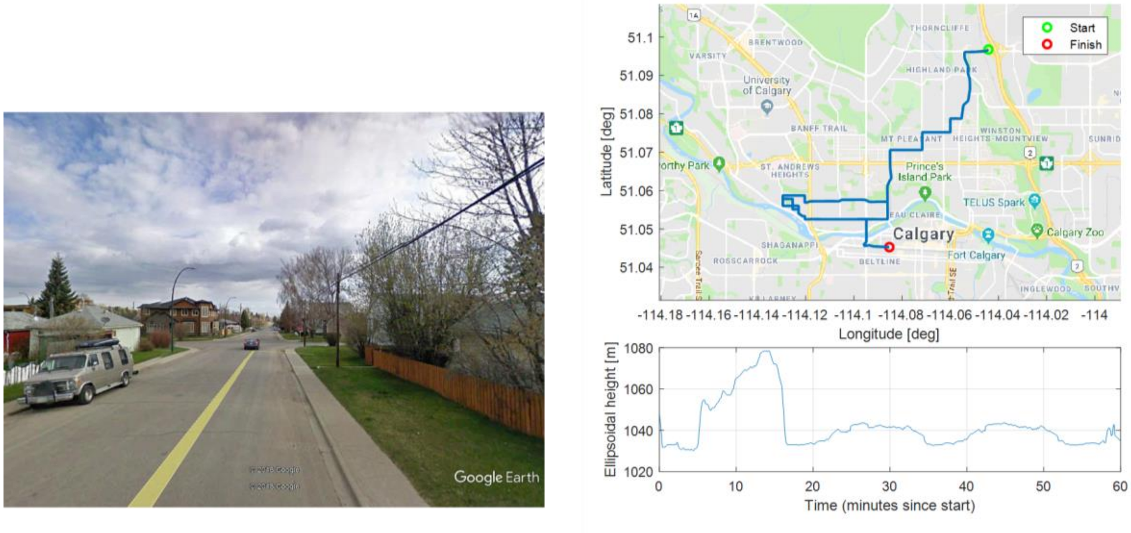


Figure 4.3. Dynamic data collected in a midtown suburban environment

The second dynamic data set travels through suburban Calgary. Large sections of the environment are fairly benign, but there are points where there are partial or full measurement outages.

- Receiver: NovAtel OEM 7500
- 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

4.4 Dynamic flight data

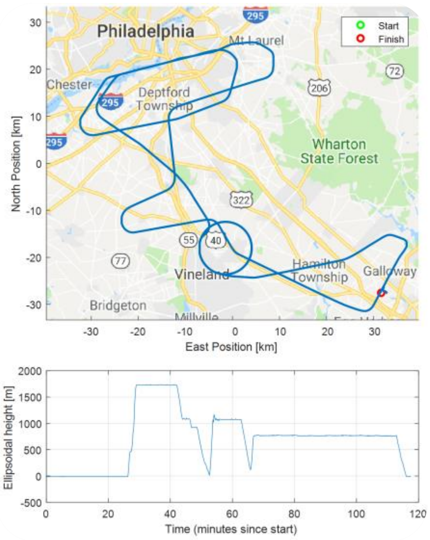


Figure 4.4. Dynamic flight data

The third dynamic 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

4.5 PPP external inputs

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

5 PROTECTION LEVEL ALGORITHM SELECTION

5.1 Parameter settings

For these simulations we used the following settings:

$$PHMI = \frac{1}{3}10^{-7} \text{ (the third accounts for all three coordinates)}$$

$$P_{FA} = \frac{1}{3}10^{-6}$$

$$P\left(H_i\right) = 10^{-5}$$

With these settings, only one-out subsets need to be formed. In section 6.2 we examine results with two-out subsets as well.

5.2 Static scenario results

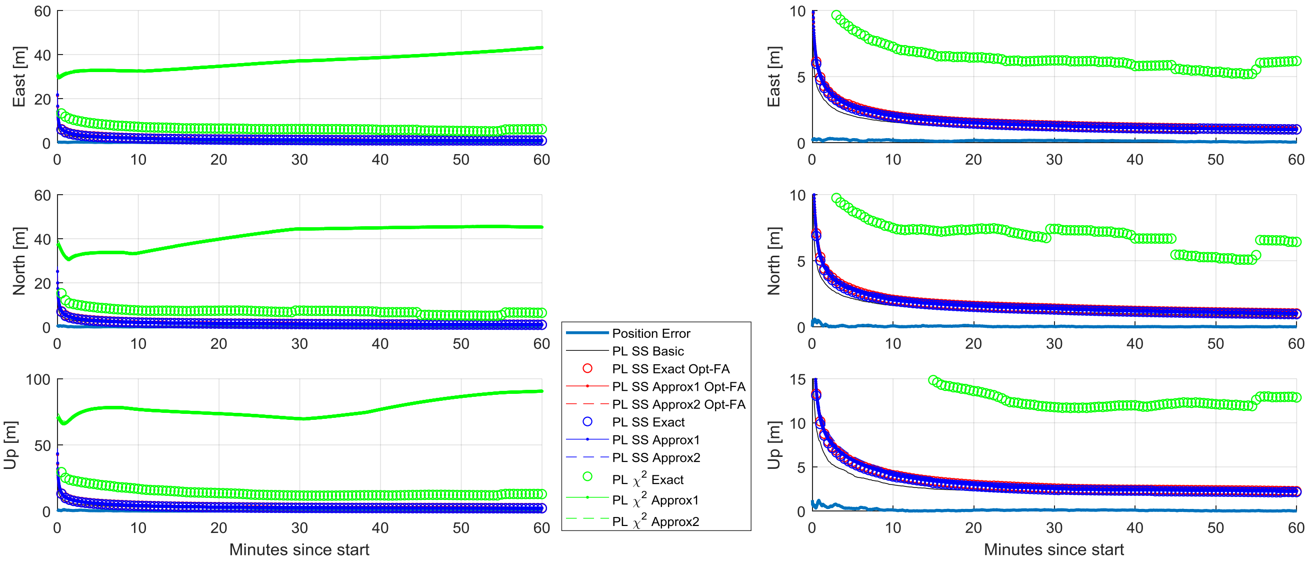


Figure 5.1 a) and b). Position error and protection levels for the static scenario- (b) is a zoom in.

5.3 Dynamic scenario results (open sky conditions)

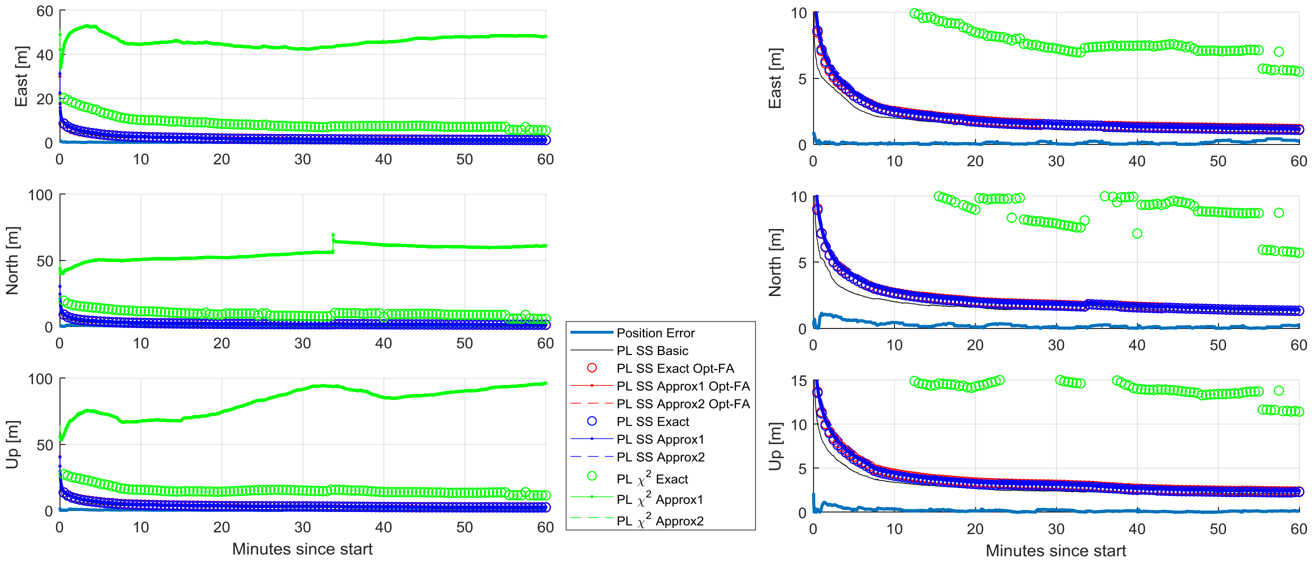


Figure 5.2 a) and b) Position error and protection levels for open sky dynamic scenario

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.

5.4 Remarks

In both scenarios, there were no detections (however, large outliers were excluded using a coarse threshold on the residuals, as indicated in section 1). 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 [10]. 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 (19), which is computationally intensive compared to (20) and (21).

Ultimately, we chose the formulation of the PL given by Equation (21), as it offers PLs close to our best algorithm with a much lower complexity and computational load.

5.6 Speed

One of the drawbacks in solution separation is the need to run parallel filters. Figure 5.5 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.

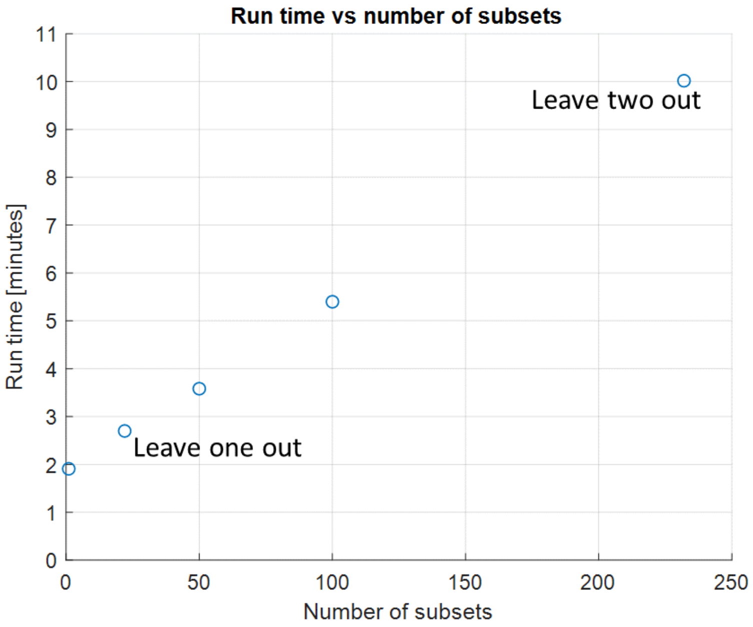


Figure 5.3 Run time as a function of the number of subsets

We observe that the run time for N filters (subsets) is much less than $N * t_{one\,filter}$. 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 naïve implementation) was achieved using by sharing modelled propagation effects across subsets, as described in section 1.

6 RESULTS

In this section, we show the results of the PPP filter and the computed protection levels for the static, driving, and flight data sets.

Static results

For these plots, we used the data described in 4.1. The filter is run in a dynamic mode just as it is for the automobile 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.

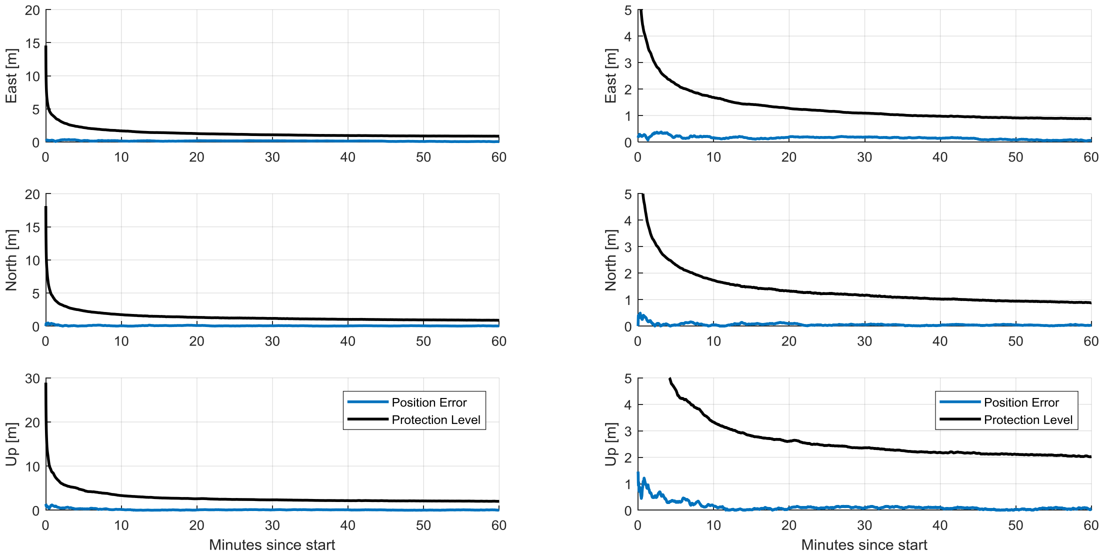


Figure 6.1 Position errors and PLs in static scenario

6.1 Dynamic results in open sky conditions

For these plots, we use the data described in Section 4.2, 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.

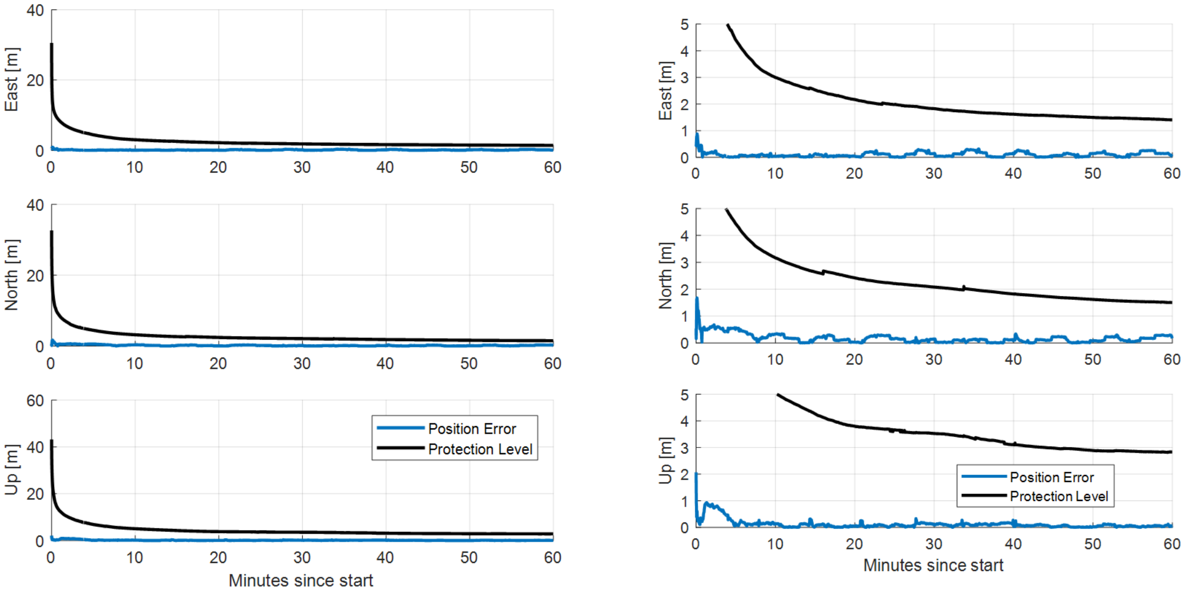


Figure 6.2 Position errors and PLs for dynamic scenario in open sky

6.2 Dynamic results in open sky conditions with two out

For this run we assumed a probability of fault large enough (10^{-4}) so that two-out subsets had to be considered as well. For this scenario, the protection levels do increase compared to the case where only one out subsets are considered (Figure 6.2) 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.

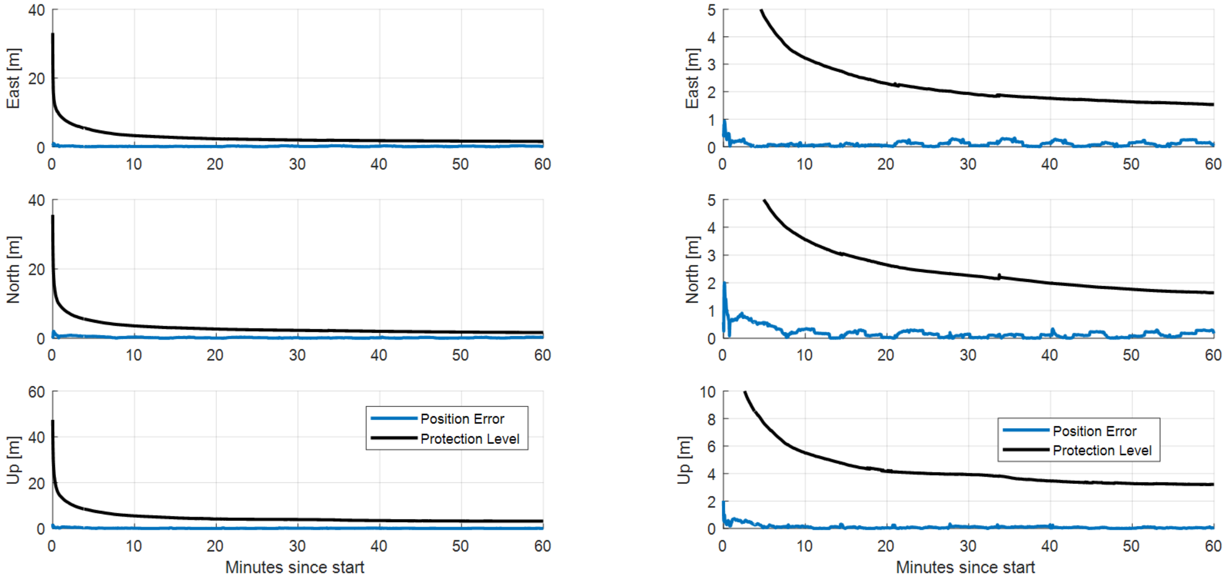


Figure 6.3 Position errors and PL for dynamic scenario in open sky conditions with two satellites out

6.3 Dynamic results in midtown suburban conditions

For these plots, we use the data described in Section 4.3, which is the midtown suburban driving scenario. This scenario contains many instances of partial or full measurement outage, which is the reason for the increased error and the numerous resets and re-convergences of the protection levels.

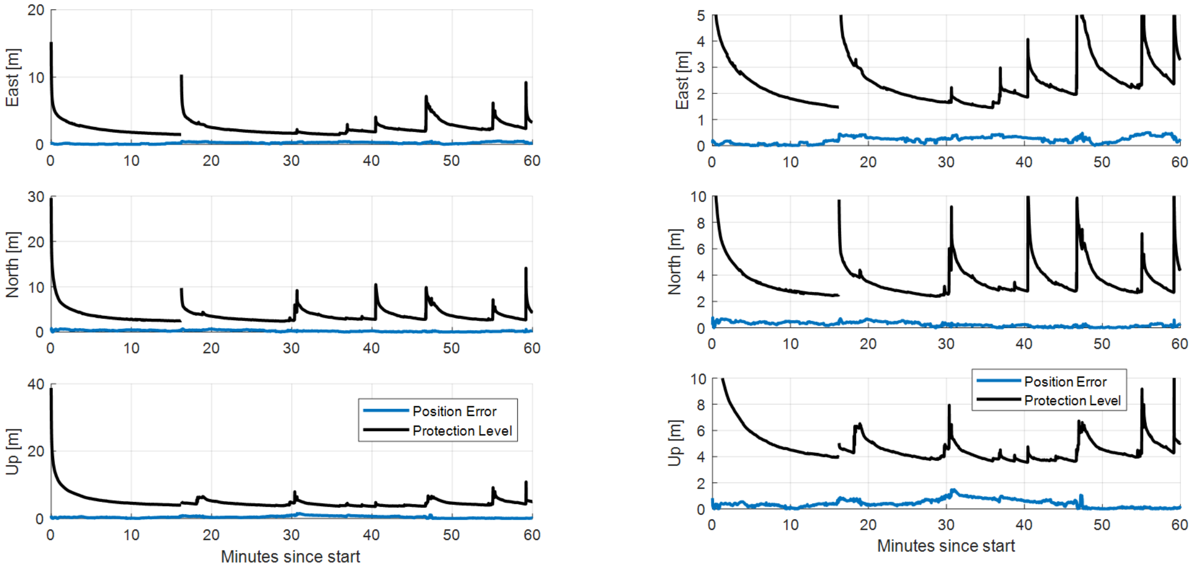


Figure 6.3 Position errors and PL for dynamic scenario in suburban conditions

6.4 Dynamic results in flight conditions

For these plots, we use the data described in Section 4.4, which is the flight scenario. The measurement environment here is generally favorable, with few measurement dropouts, leading to smooth convergence of the protection levels, which dip below one meter in the East and North directions for much of the second hour of the run.

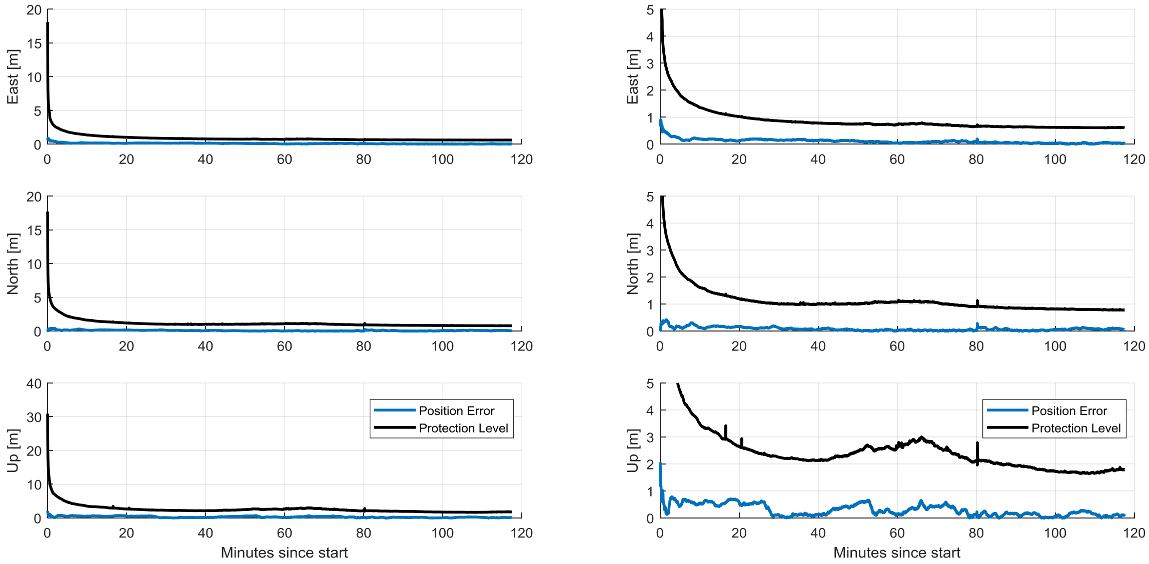


Figure 6.4 Position errors and PL for flight conditions

6.5 *Position errors under nominal conditions*

In this section, we want to make sure that: 1) our error models are reasonable, and 2) the protection levels bound the position errors in all the scenarios.

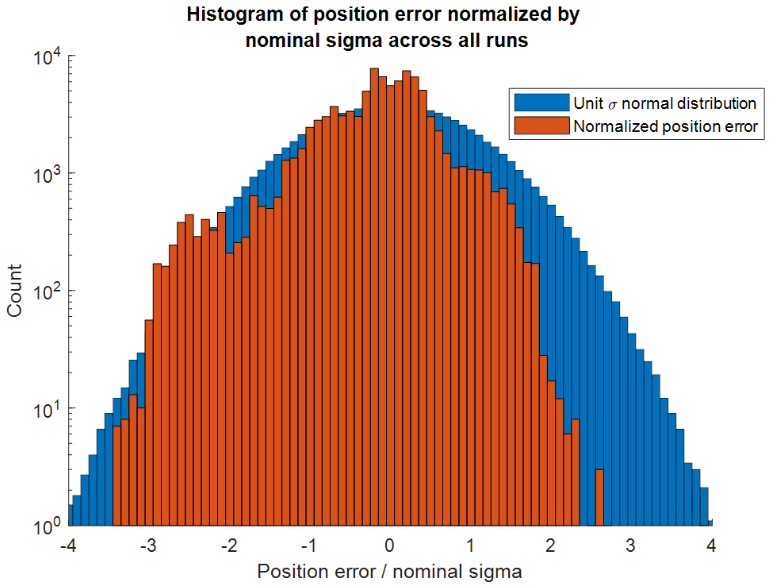


Figure 6.4 Normalized position errors compared to a normal distribution

Figure 6.4 shows that the expected distribution of position errors matches well the error models, with some conservatism, which is good. In Figure 6.5 we show the position errors normalized by the protection levels. As expected, this ratio is well below one.

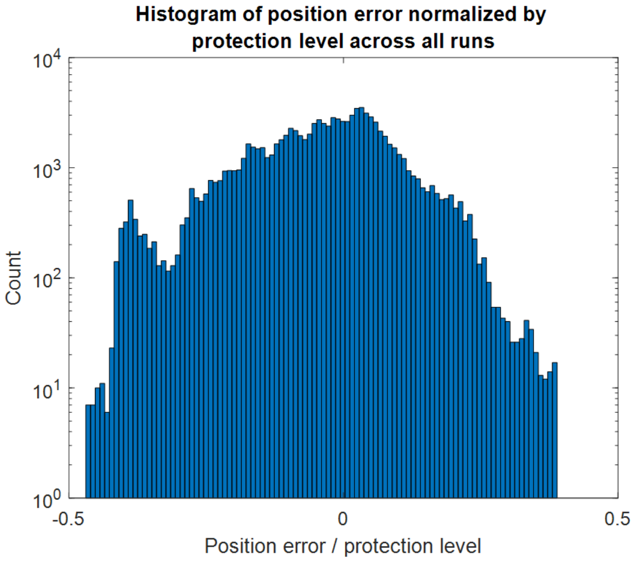


Figure 6.5 Histogram of the ratio of position error to protection level

6.6 *Exclusion function: fault injection scenario*

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.

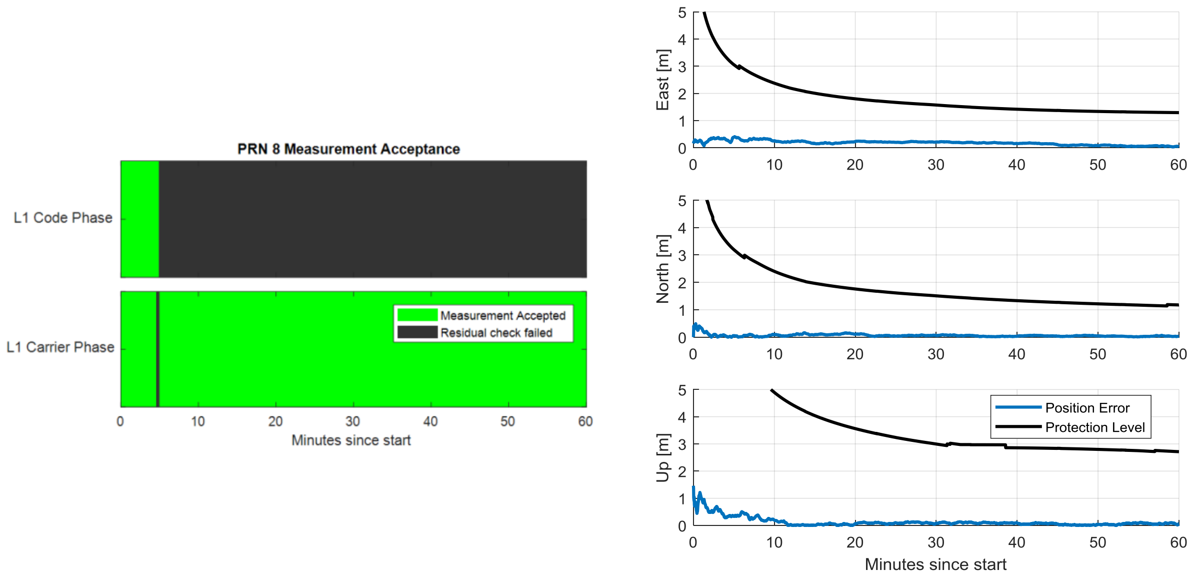


Figure 6.6. PRN 8 measurement usage (left) and position error and protection levels (right).

Whether or not the code and carrier phase measurements for PRN 8 are included in the measurement update of the EKF is shown on the left of Figure 6.6. 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 when the residual check is not enough, such as when the error slowly ramps, a second exclusion method is introduced using solution separation. 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 6.7 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.

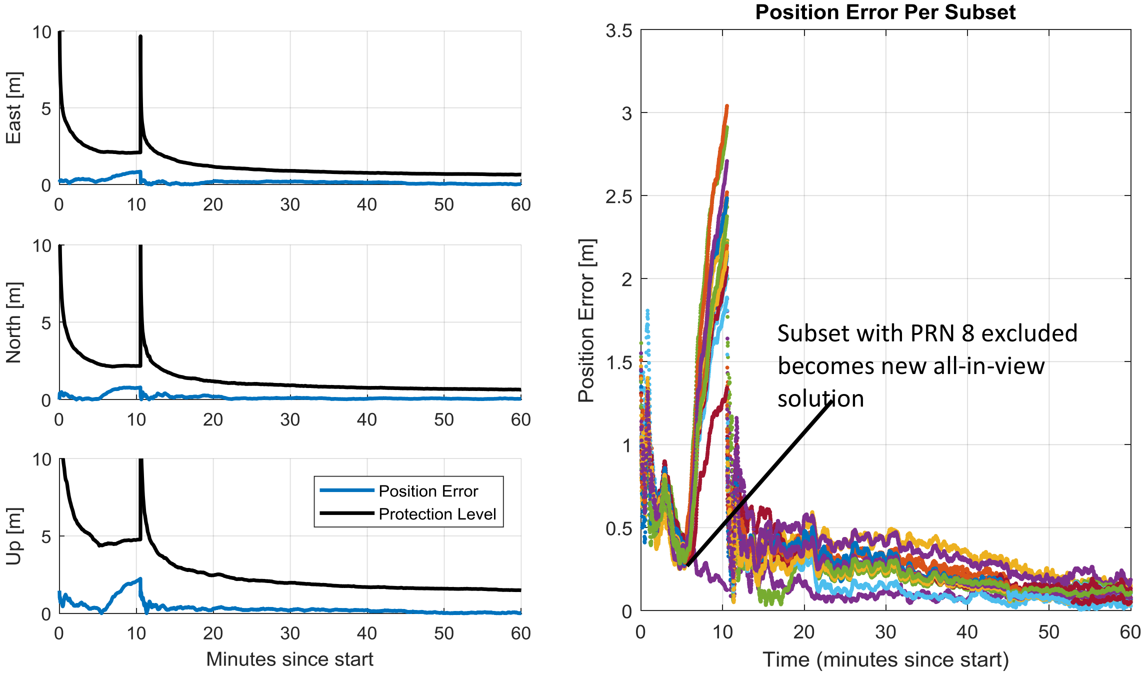


Figure 6.7. Position error and protection levels (left) and position error per subset before and after exclusion (right).

7. SUMMARY AND FUTURE WORK

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 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.

In the second phase, we have refined the PPP filter, added one scenario (suburban driving conditions), and 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.

Future work

There are still open questions:

1) Prior probabilities of fault. The validity of the protection levels computed above 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.
2) Subset management. We did not implement any of the subset management features (as for example described in in [14]). Fault consolidation in particular can significantly reduce the computational load.
3) Exclusion function. Our approach to exclusion after a solution separation test trip is sound, but it does result in a reset of all the sub-filters that causes a spike in the protection levels. It might be worthwhile exploring methods to mitigate these events. This topic is closely related to 2)
4) Further computation reduction. It is possible to reduce even further the cost of running the sub-filters by exploiting the structure of the problem.

ACKNOWLEDGEMENTS

We would like to thank Hexagon Positioning Intelligence for their partnership and for funding this research, Bill Wanner and Mike Gehringer at the FAA William J. Hughes Technical Center for providing the flight data, Stuart Riley at Trimble for lending the receiver used to collect the flight data, and the IGS for the Precise Products used in the PPP corrections.

REFERENCES

[1] Kouba, J. and Heroux, P. (2001). Precise Point Positioning Using IGS Orbit and Clock Products, GPS Solutions, vol. 5, no. 2, pp. 12-28.

[2] Madrid, P. F. Navarro, Fernández, L. Martínez, López, M. Alonso, Samper, M.D. Laínez, Merino, M.M. Romay, "PPP Integrity for Advanced Applications, Including Field Trials with Galileo, Geodetic and Low-Cost Receivers, and a Preliminary Safety Analysis," *Proceedings of the 29th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2016)*, Portland, Oregon, September 2016, pp. 3332-3354.

[3] Seepersad, G., Aggrey, J., Bisnath, S., "Do We Need Ambiguity Resolution in Multi-GNSS PPP for Accuracy or Integrity?," *Proceedings of the 30th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2017)*, Portland, Oregon, September 2017, pp. 2204-2218.

[4] de Bakker, Peter F., Tiberius, Christian C.J.M., "Single-Frequency GNSS Positioning for Assisted, Cooperative and Autonomous Driving," *Proceedings of the 30th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2017)*, Portland, Oregon, September 2017, pp. 4038-4045.

[5] Khanafseh,S., et al."GNSS Multipath Error Modeling for Automotive Applications" *Proceedings of the 30th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2018)*, Miami, Florida, September 2018.

[6] Blanch, J., Walter, T., Enge, P., Lee, Y., Pervan, B., Rippl, M., Spletter, A., Kropp, V., "Baseline Advanced RAIM User A6gorithm and Possible Improvements," IEEE Transactions on Aerospace and Electronic Systems, Volume 51, No. 1, January 2015.

[7] Blanch, J., Walter, T., Enge, P. "RAIM with Optimal Integrity and Continuity Allocations Under Multiple failures," IEEE Transactions on Aerospace and Electronic Systems Vol. 46, No. 3, July 2010.

[8] Working Group C, ARAIM Technical Subgroup, Milestone 3 Report, February 26, 2016. Available at:

http://www.gps.gov/policy/cooperation/europe/2016/working-group-c/

http://ec.europa.eu/growth/tools-databases/newsroom/cf/itemdetail.cfm?item_id=8690

[9] Joerger, Mathieu, Chan, Fang-Cheng, Pervan, Boris, "Solution Separation Versus Residual-Based RAIM", *NAVIGATION, Journal of The Institute of Navigation*, Vol. 61, No. 4, Winter 2014, pp. 273-291.

[10] Blanch, Juan, Walter, Todd, Enge, Per, "Theoretical Results on the Optimal Detection Statistics for Autonomous Integrity Monitoring", NAVIGATION, Journal of The Institute of Navigation, Vol. 64, No. 1, Spring 2017, pp. 123-137.

[11] B. DeCleene, "Defining Pseudorange Integrity - Overbounding," *Proceedings of the 13th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2000)*, Salt Lake City, UT, September 2000, pp. 1916-1924

[12] J. Rife, S. Pullen, B. Pervan, and P. Enge. Paired Overbounding for Nonideal LAAS and WAAS Error Distributions. IEEE Transactions on Aerospace and Electronic Systems, 2006, 42, 4, 1386 -1395

[13] J. Blanch, T. Walter, and P. Enge. "A MATLAB Toolset to Determine Strict Gaussian Bounding Distributions of a Sample Distribution," *Proceedings of the 30th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2017)*, Portland, Oregon, September 2017, pp. 4236-4247.

[14] Blanch, Juan, Walter, Todd, Enge, Per, "Fixed Subset Selection to Reduce Advanced RAIM Complexity," *Proceedings of the 2018 International Technical Meeting of The Institute of Navigation*, Reston, Virginia, January 2018, pp. 88-98.

APPENDIX A

Formulas with innovation:

$$\chi^2\left(t\right)=\chi^2\left(t-1\right)+\left(y_t-G_t\hat{x}_{t|t-1}\right)^T\left(W_t-W_tG_tP_{t|t}G_t^TW_t\right)\left(y_t-G_t\hat{x}_{t|t-1}\right) \qquad (30)$$

Or

$$\chi^2\left(t\right)=\chi^2\left(t-1\right)+\left(y_t-G_t\hat{x}_{t|t-1}\right)^T\left(W_t^{-1}+G_tP_{t|t-1}G_t^T{}_t\right)^{-1}\left(y_t-G_t\hat{x}_{t|t-1}\right) \qquad (31)$$

Where:

$$W_t=R_t^{-1}$$

This is true because we have the identity (see proof below):

$$W_t-W_tG_tP_{t|t}G_t^TW_t=\left(W_t^{-1}+G_tP_{t|t-1}G_t^T\right)^{-1} \qquad (32)$$

Formulas with residuals

$$\chi^2(t)=\chi^2(t-1)+\left(y_t-G_t\hat{x}_{t|t}\right)^T W_t\left(W_t-W_tG_tP_{t|t}G_t{}^TW_t\right)^{-1}W_t\left(y_t-G_t\hat{x}_{t|t}\right) \tag{33}$$

or

$$\chi^2(t)=\chi^2(t-1)+\left(y_t-G_t\hat{x}_{t|t}\right)^T\left(W_t+W_tG_tP_{t|t-1}G_t{}^TW_t\right)\left(y_t-G_t\hat{x}_{t|t}\right) \tag{34}$$

Proof for Equations (33) and (34) (from innovation formulas)

The formula for the state update is:

$$\hat{x}_{t|t}=\hat{x}_{t|t-1}+P_{t|t}G_t^TW_t\left(y_t-G_t\hat{x}_{t|t-1}\right) \tag{35}$$

Which we can write as:

$$\hat{x}_{t|t}=\left(I-P_{t|t}G_t^TW_tG_t\right)\hat{x}_{t|t-1}+P_{t|t}G_t^TW_ty_t \tag{36}$$

We invert this equation:

$$\hat{x}_{t|t-1}=\left(I-P_{t|t}G_t^TW_tG_t\right)^{-1}\left(\hat{x}_{t|t}-P_{t|t}G_t^TW_ty_t\right) \tag{37}$$

We now replace $\hat{x}_{t|t-1}$ by the above expression in Equation (30). We get:

$$\chi^2(t)=\chi^2(t-1)+$$
$$\left(y_t-G_t\left(I-P_{t|t}G_t^TW_tG_t\right)^{-1}\left(\hat{x}_{t|t}-P_{t|t}G_t^TW_ty_t\right)\right)^T\left(W_t-W_tG_tP_{t|t}G_t{}^TW_t\right)\left(y_t-G_t\left(I-P_{t|t}G_t^TW_tG_t\right)^{-1}\left(\hat{x}_{t|t}-P_{t|t}G_t^TW_ty_t\right)\right) \tag{38}$$

For the term highlighted in blue we have:

$$\left(W_t-W_tG_tP_{t|t}G_t{}^TW_t\right)\left(y_t-G_t\left(I-P_{t|t}G_t^TW_tG_t\right)^{-1}\left(\hat{x}_{t|t}-P_{t|t}G_t^TW_ty_t\right)\right)=$$
$$W_t\left(y_t-G_t\hat{x}_{t|t}\right) \tag{39}$$

We can this by developing this term in two contributions. First term:

$$\left(W_t-W_tG_tP_{t|t}G_t{}^TW_t\right)y_t$$

Second term:

$$-\left(W_t - W_t G_t P_{t|t} G_t{}^T W_t\right) G_t \left(I - P_{t|t} G_t^T W_t G_t\right)^{-1} \left(\hat{x}_{t|t} - P_{t|t} G_t^T W_t y_t\right) =$$

$$-W_t G_t \left(I - P_{t|t} G_t^T W_t G_t\right)\left(I - P_{t|t} G_t^T W_t G_t\right)^{-1} \left(\hat{x}_{t|t} - P_{t|t} G_t^T W_t y_t\right) =$$

$$-W_t G_t \left(\hat{x}_{t|t} - P_{t|t} G_t^T W_t y_t\right)$$

For the term highlighted in blue we have:

$$\left(y_t - G_t \left(I - P_{t|t} G_t^T W_t G_t\right)^{-1} \left(\hat{x}_{t|t} - P_{t|t} G_t^T W_t y_t\right)\right)^T =$$

$$\left(\left(W_t - W_t G_t P_{t|t} G_t{}^T W_t\right)\left(y_t - G_t \left(I - P_{t|t} G_t^T W_t G_t\right)^{-1} \left(\hat{x}_{t|t} - P_{t|t} G_t^T W_t y_t\right)\right)\right)^T \left(W_t - W_t G_t P_{t|t} G_t{}^T W_t\right)^{-1} \quad (40)$$

$$= \left(W_t \left(y_t - G_t \hat{x}_{t|t}\right)\right)^T \left(W_t - W_t G_t P_{t|t} G_t{}^T W_t\right)^{-1}$$

Putting Equations (40) and (39) in (38), we get:

$$\chi^2(t) = \chi^2(t-1) + \left(y_t - G_t \hat{x}_{t|t}\right)^T W_t \left(W_t - W_t G_t P_{t|t} G_t{}^T W_t\right)^{-1} W_t \left(y_t - G_t \hat{x}_{t|t}\right) \quad (41)$$

We also have:

$$\chi^2(t) = \chi^2(t-1) + \left(y_t - G_t \hat{x}_{t|t}\right)^T \left(W_t + W_t G_t P_{t|t-1} G_t{}^T W_t\right)\left(y_t - G_t \hat{x}_{t|t}\right) \quad (42)$$

This last equation comes from the identity:

$$W_t \left(W_t - W_t G_t P_{t|t} G_t{}^T W_t\right)^{-1} W_t = W_t + W_t G_t P_{t|t-1} G_t{}^T W_t \quad (43)$$

Which is directly derived from Equation (32)

Proof for Equation (32)

The matrix inversion lemma gives us:

$$\left(A - BD^{-1}C\right)^{-1} = A^{-1} + A^{-1}B\left(D - CA^{-1}B\right)^{-1} CA^{-1}$$

We apply it to:

$$A = W_t \quad B = W_t G_t \quad C = G_t{}^T W \quad D = P_{t|t}{}^{-1}$$

And get:

$$\left(W_t - W_t G_t P_{t|t} G_t{}^T W_t\right)^{-1} = W_t^{-1} + W_t^{-1} W_t G_t \left(P_{t|t} - G_t{}^T W G_t\right)^{-1} P_{t|t-1} G_t{}^T W_t W_t^{-1} \quad (44)$$

By the definition of the posterior covariance, we have:

$$P_{t|t} = \left(P_{t|t-1}{}^{-1} + G_t{}^{T} W G_t\right)^{-1} \tag{45}$$

Which using (45) provides (32)

Proof of Equation (30)

Notations for the batch process:

$\bar{G}_t$ observation matrix including all measurements up to *t* and propagation equations up to *t*+1

$\bar{y}_t$ measurements up to t (including observations accounting for the propagation)

$\bar{W}_t$ weighting matrix of all measurements

$\bar{P}_{t|t-1} = \left(\bar{G}_{t-1}^{T}\bar{W}_{t-1}\bar{G}_{t-1}\right)^{-1}$

$\tilde{G}_t$ observation matrix of the measurements added at time t (but with columns accounting for all states)

The chi-square of the measurement residuals up to time t-1 (including the propagation equations to t) is given by:

$$\begin{aligned}\chi^2\left(t-1\right) &= \bar{y}_{t-1}^{T}\left(\bar{W}_{t-1} - \bar{W}_{t-1}\bar{G}_{t-1}\left(\bar{G}_{t-1}^{T}\bar{W}_{t-1}\bar{G}_{t-1}\right)^{-1}\bar{G}_{t-1}^{T}\bar{W}_{t-1}\right)\bar{y}_{t-1} \\ &= \bar{y}_{t-1}^{T}\left(\bar{W}_{t-1} - \bar{W}_{t-1}\bar{G}_{t-1}\bar{P}_{t|t-1}\bar{G}_{t-1}^{T}\bar{W}_{t-1}\right)\bar{y}_{t-1}\end{aligned} \tag{46}$$

(We will later on show that adding the propagation equations does not modify the residuals (because we add as many states as we add measurements))

The batch chi-square at time *t* uses the same formula but where have added the new measurements:

$$\chi^2\left(t\right) = \begin{bmatrix}\bar{y}_{t-1}^{T} & y_t^{T}\end{bmatrix}\left(\begin{bmatrix}\bar{W}_{t-1} & 0 \\ 0 & W_t\end{bmatrix} - \begin{bmatrix}\bar{W}_{t-1}\bar{G}_{t-1} \\ W_t\tilde{G}_t\end{bmatrix}\bar{P}_{t|t}\begin{bmatrix}\bar{G}_{t-1}^{T}\bar{W}_{t-1} & \tilde{G}_t^{T}W_t\end{bmatrix}\right)\begin{bmatrix}\bar{y}_{t-1} \\ y_t\end{bmatrix}\bar{y}_{t-1} \tag{47}$$

where:

$$\begin{aligned}\bar{P}_{t|t} &= \left(\begin{bmatrix}\bar{G}_{t-1}^{T} & \tilde{G}_t^{T}\end{bmatrix}\begin{bmatrix}\bar{W}_{t-1} & 0 \\ 0 & W_t\end{bmatrix}\begin{bmatrix}\bar{G}_{t-1} \\ \tilde{G}_t\end{bmatrix}\right)^{-1} \\ &= \left(\bar{G}_{t-1}^{T}\bar{W}_{t-1}\bar{G}_{t-1} + \tilde{G}_t^{T}W_t\tilde{G}_t\right)^{-1} \\ &= \left(\bar{P}_{t|t-1}{}^{-1} + \tilde{G}_t^{T}W_t\tilde{G}_t\right)^{-1}\end{aligned} \tag{48}$$

We develop Equation (47):

$$\chi^2(t) = \bar{y}_{t-1}^T \bar{W}_{t-1} \bar{y}_{t-1} + y_t^{\,T} W_t y_t - \begin{bmatrix} \bar{y}_{t-1}^T \bar{W}_{t-1} \bar{G}_{t-1} \\ y_t^T W_t \tilde{G}_t \end{bmatrix} \bar{P}_{t|t} \begin{bmatrix} \bar{G}_{t-1}^T \bar{W}_{t-1} \bar{y}_{t-1} & \tilde{G}_t^T W_t y_t \end{bmatrix} \tag{49}$$

Developing further, we get:

$$\begin{aligned} \chi^2(t) &= \bar{y}_{t-1}^T \bar{W}_{t-1} \bar{y}_{t-1} + \bar{y}_{t-1}^T W_t y_t \\ &- \bar{y}_{t-1}^T \bar{W}_{t-1} \bar{G}_{t-1} \bar{P}_{t|t} \bar{G}_{t-1}^T \bar{W}_{t-1} \bar{y}_{t-1} \\ &- 2 y_t^T W_t \tilde{G}_t \bar{P}_{t|t} \bar{G}_{t-1}^T \bar{W}_{t-1} \bar{y}_{t-1} \\ &- y_t^T W_t \tilde{G}_t \bar{P}_{t|t} \tilde{G}_t^T W_t y_t \end{aligned} \tag{50}$$

From the definition of the error covariance (Equation (49)), we have:

$$\begin{aligned} \bar{P}_{t|t} &= \bar{P}_{t|t} \bar{P}_{t|t-1}^{\ -1} \bar{P}_{t|t-1} = \bar{P}_{t|t} \left(\bar{P}_{t|t}^{\ -1} - \tilde{G}_t^T W_t \tilde{G}_t \right) \bar{P}_{t|t-1} = \left(I - \bar{P}_{t|t} \tilde{G}_t^T W_t \tilde{G}_t \right) \bar{P}_{t|t-1} \\ &= \bar{P}_{t|t-1} \left(I - \tilde{G}_t^T W_t \tilde{G}_t \bar{P}_{t|t} \right) \end{aligned} \tag{51}$$

We put this expression in the third line of (50)

$$\begin{aligned} & 2 y_t^T W_t \tilde{G}_t \bar{P}_{t|t} \bar{G}_{t-1}^T \bar{W}_{t-1} \bar{y}_{t-1} = \\ & 2 y_t^T W_t \tilde{G}_t \left(I - \bar{P}_{t|t} \tilde{G}_t^T W_t \tilde{G}_t \right) \bar{P}_{t|t-1} \bar{G}_{t-1}^T \bar{W}_{t-1} \bar{y}_{t-1} = \\ & 2 y_t^T \left(W_t - W_t \tilde{G}_t \bar{P}_{t|t} \tilde{G}_t^T W_t \right) \tilde{G}_t \bar{P}_{t|t-1} \bar{G}_{t-1}^T \bar{W}_{t-1} \bar{y}_{t-1} = \\ & 2 y_t^T \left(W_t - W_t \tilde{G}_t \bar{P}_{t|t} \tilde{G}_t^T W_t \right) \tilde{G}_t \bar{P}_{t|t-1} \bar{G}_{t-1}^T \bar{W}_{t-1} \bar{y}_{t-1} = \\ & 2 y_t^T \left(W_t - W_t \tilde{G}_t \bar{P}_{t|t} \tilde{G}_t^T W_t \right) \tilde{G}_t \hat{X}_{t-1} \end{aligned} \tag{52}$$

Where:

$$\hat{X}_{t-1} = \bar{P}_{t|t-1} \bar{G}_{t-1}^T \bar{W}_{t-1} \bar{y}_{t-1}$$

is the batch estimate at *t*-1.

We now put the expression (51) in the second line of equation (50) and we get:

$$\begin{aligned} & \bar{y}_{t-1}^T \bar{W}_{t-1} \bar{G}_{t-1} \bar{P}_{t|t} \bar{G}_{t-1}^T \bar{W}_{t-1} \bar{y}_{t-1} = \bar{y}_{t-1}^T \bar{W}_{t-1} \bar{G}_{t-1} \left(I - \bar{P}_{t|t} \tilde{G}_t^T W_t \tilde{G}_t \right) \bar{P}_{t|t-1} \bar{G}_{t-1}^T \bar{W}_{t-1} \bar{y}_{t-1} \\ & = \bar{y}_{t-1}^T \bar{W}_{t-1} \bar{G}_{t-1} \bar{P}_{t|t-1} \bar{G}_{t-1}^T \bar{W}_{t-1} \bar{y}_{t-1} - \bar{y}_{t-1}^T \bar{W}_{t-1} \bar{G}_{t-1} \bar{P}_{t|t} \tilde{G}_t^T W_t \tilde{G}_t \bar{P}_{t|t-1} \bar{G}_{t-1}^T \bar{W}_{t-1} \bar{y}_{t-1} \end{aligned} \tag{53}$$

We use it again in the second term of the above expression:

$$\begin{aligned}&\bar{y}_{t-1}^{T}\bar{W}_{t-1}\bar{G}_{t-1}\bar{P}_{t|t}\bar{G}_{t-1}^{T}\bar{W}_{t-1}\bar{y}_{t-1} = \\ &= \bar{y}_{t-1}^{T}\bar{W}_{t-1}\bar{G}_{t-1}\bar{P}_{t|t-1}\bar{G}_{t-1}^{T}\bar{W}_{t-1}\bar{y}_{t-1} - \bar{y}_{t-1}^{T}\bar{W}_{t-1}\bar{G}_{t-1}\bar{P}_{t|t-1}\left(I - \tilde{G}_t^{T}W_t\tilde{G}_t\bar{P}_{t|t}\right)\tilde{G}_t^{T}W_t\tilde{G}_t\bar{P}_{t|t-1}\bar{G}_{t-1}^{T}\bar{W}_{t-1}\bar{y}_{t-1} \\ &= \bar{y}_{t-1}^{T}\bar{W}_{t-1}\bar{G}_{t-1}\bar{P}_{t|t-1}\bar{G}_{t-1}^{T}\bar{W}_{t-1}\bar{y}_{t-1} - \bar{y}_{t-1}^{T}\bar{W}_{t-1}\bar{G}_{t-1}\bar{P}_{t|t-1}\tilde{G}_t^{T}\left(W_t - W_t\tilde{G}_t\bar{P}_{t|t}\tilde{G}_t^{T}W_t\right)\tilde{G}_t\bar{P}_{t|t-1}\bar{G}_{t-1}^{T}\bar{W}_{t-1}\bar{y}_{t-1}\end{aligned} \tag{54}$$

We therefore have:

$$\bar{y}_{t-1}^{T}\bar{W}_{t-1}\bar{G}_{t-1}\bar{P}_{t|t}\bar{G}_{t-1}^{T}\bar{W}_{t-1}\bar{y}_{t-1} = \bar{y}_{t-1}^{T}\bar{W}_{t-1}\bar{G}_{t-1}\bar{P}_{t|t-1}\bar{G}_{t-1}^{T}\bar{W}_{t-1}\bar{y}_{t-1} - \hat{X}_{t-1}^{T}\tilde{G}_t^{T}\left(W_t - W_t\tilde{G}_t\bar{P}_{t|t}\tilde{G}_t^{T}W_t\right)\tilde{G}_t\hat{X}_{t-1} \tag{55}$$

Now we insert (52) and (55) in (50), and we get:

$$\begin{aligned}&\chi^2(t) = \bar{y}_{t-1}^{T}\bar{W}_{t-1}\bar{y}_{t-1} + y_t^{\,T}W_t y_t \\ &-\bar{y}_{t-1}^{T}\bar{W}_{t-1}\bar{G}_{t-1}\bar{P}_{t|t-1}\bar{G}_{t-1}^{T}\bar{W}_{t-1}\bar{y}_{t-1} + \hat{X}_{t-1}^{T}\tilde{G}_t^{T}\left(W_t - W_t\tilde{G}_t\bar{P}_{t|t}\tilde{G}_t^{T}W_t\right)\tilde{G}_t\hat{X}_{t-1} \\ &-2y_t^{T}\left(W_t - W_t\tilde{G}_t\bar{P}_{t|t}\tilde{G}_t^{T}W_t\right)\tilde{G}_t\hat{X}_{t-1} \\ &-y_t^{T}W_t\tilde{G}_t\bar{P}_{t|t}\tilde{G}_t^{T}W_t y_t\end{aligned} \tag{56}$$

Rearranging the terms, we get:

$$\begin{aligned}&\chi^2(t) = \bar{y}_{t-1}^{T}\bar{W}_{t-1}\bar{y}_{t-1} - \bar{y}_{t-1}^{T}\bar{W}_{t-1}\bar{G}_{t-1}\bar{P}_{t|t-1}\bar{G}_{t-1}^{T}\bar{W}_{t-1}\bar{y}_{t-1} \\ &+y_t^{\,T}\left(W_t - W_t\tilde{G}_t\bar{P}_{t|t}\tilde{G}_t^{T}W_t\right)_t y_t \\ &+\hat{X}_{t-1}^{T}\tilde{G}_t^{T}\left(W_t - W_t\tilde{G}_t\bar{P}_{t|t}\tilde{G}_t^{T}W_t\right)\tilde{G}_t\hat{X}_{t-1} \\ &-2y_t^{T}\left(W_t - W_t\tilde{G}_t\bar{P}_{t|t}\tilde{G}_t^{T}W_t\right)\tilde{G}_t\hat{X}_{t-1} \\ &= \chi^2(t-1) + \left(y_t - \tilde{G}_t^{T}\hat{X}_{t-1}\right)^{T}\left(W_t - W_t\tilde{G}_t\bar{P}_{t|t}\tilde{G}_t^{T}W_t\right)\left(y_t - \tilde{G}_t^{T}\hat{X}_{t-1}\right)\end{aligned} \tag{57}$$

By noticing that $\tilde{G}_t^{T}\hat{X}_{t-1} = G_t\hat{x}_{t|t-1}$ and $\tilde{G}_t\bar{P}_{t|t}\tilde{G}_t^{T} = G_tP_{t|t}G_t^{\,T}$ we get Equation (30)