Reducing Computational Load in Solution Separation for Kalman Filters and an Application to PPP Integrity

Juan Blanch, Kaz Gunning, Todd Walter. *Stanford University*

Lance De Groot, Laura Norman. *Hexagon Positioning Intelligence*

ABSTRACT

This paper investigates two techniques to reduce the computational load of running multiple fault tolerant Kalman filters in order to provide integrity. These approaches are then exploited in the implementation of a solution separation integrity monitoring algorithm in a PPP Kalman filter solution. We evaluate the techniques using GNSS data collected in static and driving conditions. In our scenarios, these techniques lead to computational load reductions of at least 70% at the expense of protection level degradations of about 50%.

INTRODUCTION

Until recently, Precise Point Positioning (PPP) techniques [1] have mostly been used to provide high accuracy. There is a growing interest in translating the benefits of PPP to integrity and enabling its application to safety critical applications in rail, automotive, maritime, and even air navigation [2], [3], [4], [5]. In [5], we demonstrated how techniques developed for aviation applied to PPP can produce meter-level protection levels in automotive and aviation scenarios. This was achieved by implementing an integrity monitoring algorithm based on solution separation, akin to the one used to analyze Advanced RAIM performance [6], to the PPP Kalman filter solution.

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. In addition to their optimality properties [7], solution separation algorithms offer a straightforward proof of integrity, and good performance [5]. However, they can also be expensive in terms of memory and processing time, because they require the receiver to compute a bank of filters (or a process computationally equivalent to a bank of filters, as in [11]). In the worst case, the computational load will be proportional to the number of filters.

In [5] we showed that it was possible to dramatically reduce the cost of running the bank of filters: depending on the filter complexity (that is, the number of estimated states), we could run 20 to 50 additional filters for the cost of one. This was obtained by exploiting the fact that, in PPP, many of the elements in the computation (error models, corrections, etc) are common to the all filters, so that it is sufficient to compute them once for the all-in-view filter. Also, all the measurements are linearized with respect to the all-in-view position solution, which further simplifies the subset solution filters.

The goal of this paper is to introduce and investigate techniques to reduce even further the cost of the solution separation for Kalman filter solutions. When the number of states is large (larger than 50), which is the expectation in a PPP multi-frequency user algorithm, there are at least two steps that are computationally expensive: the determination of the Kalman gain, and the determination of the new error covariance.

The first technique under investigation consists in using a set of suboptimal filters for the subset solutions (instead of the optimal filter), where the Kalman gain for each subfilter is derived from the all-in-view and does not require a full matrix

inversion. The second technique that we will evaluate is the consolidation of faults into a few subsets. This is another type of suboptimal subset solution that has already been evaluated for snapshot solutions in Advanced RAIM [8], [9].

For the experimental evaluation, we will use our sequential PPP filter implementation [5], which is based on a simple extended Kalman filter with estimated parameters comprising the receiver position, clock biases for each constellation in use, a tropospheric delay, and float ambiguities for each tracked carrier phase. Dual-frequency measurements are incorporated from GPS, GLONASS, Galileo, and BeiDou. The precise orbit and clock estimates are drawn from the IGS MGEX analysis centers.

SUBOPTIMAL SUBSET SOLUTIONS: FIRST APPROACH

For the approach outlined here, we only consider the measurement update step of the Kalman filter. For the all-in-view filter (indicated by the index 0), we have the following Kalman filter equations:

$$\hat{x}_{t+1|t+1}^{(0)} = \hat{x}_{t+1|t}^{(0)} + C_{t+1|t+1}^{(0)} G^T W \left(y_{t+1} - G\hat{x}_{t+1|t}^{(0)} \right) \tag{1}$$

Where $\hat{x}_{t+1|t+1}^{(0)}$ is the a posteriori state estimate, $\hat{x}_{t+1|t}^{(0)}$ is the a priori estimate, *G* is the observation matrix, *W* is the inverse of the measurement noise matrix, *y* is the vector of measurements, and $C_{t+1|t+1}^{(0)}$ is the error covariance of the a posteriori state estimate. We have:

$$C_{t+1|t+1}^{(0)} = \left(\left(C_{t+1|t}^{(0)} \right)^{-1} + G^T W G \right)^{-1} \tag{2}$$

Where $\left(C_{t+1|t}^{(0)} \right)^{-1}$ is the error covariance of the a priori estimate.

Subset filter solutions

In this paper we will consider probability faults of 10^{-5} and 10^{-4} per hour. For an integrity of 10^{-7} per hour, this means that we need to compute the solution separation statistic of all one out subsets in the case of 10^{-5} and two out subsets for 10^{-4} [8]. The subset filter Kalman filter equations (indexed by *k*) are similar to the all-in-view ones:

$$\hat{x}_{t+1|t+1}^{(k)} = \hat{x}_{t+1|t}^{(k)} + C_{t+1|t+1}^{(k)} G^{(k)T} W^{(k)} \left(y_{t+1} - G^{(k)} \hat{x}_{t+1|t}^{(k)} \right) \tag{3}$$

$$C_{t+1|t+1}^{(k)} = \left(\left(C_{t+1|t}^{(k)} \right)^{-1} + G^{(k)T} W^{(k)} G^{(k)} \right)^{-1} \tag{4}$$

The only difference with respect to the all-in-view is that we only use a subset of the available measurements to update the state estimate. One of the most onerous steps in this process is the computation of the covariance as written in Equation (4) . As can be seen, we need at least two matrix inversions, where both matrices are *n* by *n* with *n* being close to a 100. (We note that the geometry matrix is often as large, so a matrix update formula will not substantially reduce the computational load).

The first approach consists in using a suboptimal filter for $\hat{x}_{t+1|t+1}^{(k)}$ instead of the optimal one defined above. More precisely, we define it as follows:

$$\hat{x}_{t+1|t+1}^{(k)} = \hat{x}_{t+1|t}^{(k)} + \Sigma_{t+1|t+1}^{(k)} G^{(k)T} W^{(k)} \left(y^{(k)} - G^{(k)} \hat{x}_{t+1|t}^{(k)} \right) \tag{5}$$

Where the matrix $\Sigma_{t+1|t+1}^{(k)}$ is no longer given by Equation (4). Instead, we attempt to find a matrix that will result in a reasonable estimator but that is cheaper to compute. One possible approach is to compute this matrix as if the prior of the estimated states was given by the prior of the all-in-view filter:

$$\Sigma_{t+1|t+1}^{(k)} = \left(\left(C_{t+1|t}^{(0)} \right)^{-1} + G^{(k)T} W^{(k)} G^{(k)} \right)^{-1} \tag{6}$$

The advantage is that this matrix can be obtained without a full matrix inversion. We have:

$$\Sigma_{t+1|t+1}^{(k)} = \left(\left(C_{t+1|t}^{(0)} \right)^{-1} + G^{T} W G + G^{(k)T} W^{(k)} G^{(k)} - G^{T} W G \right)^{-1} \tag{7}$$

In most cases, the rank of the matrix $G^{(k)T} W^{(k)} G^{(k)} - G^{T} W G$ is considerably smaller than the rank of $\left(C_{t+1|t}^{(0)} \right)^{-1} + G^{T} W G$. For example, in the one out case of our PPP filter, the rank of this matrix is 4. We can write:

$$G^{T} W G - G^{(k)T} W^{(k)} G^{(k)} = \bar{G}^{(k)T} \bar{W}^{(k)} \bar{G}^{(k)} \tag{8}$$

$$\Sigma_{t+1|t+1}^{(k)} = \left(\left(C_{t+1|t}^{(0)} \right)^{-1} + G^{T} W G - \bar{G}^{(k)T} \bar{W}^{(k)} \bar{G}^{(k)} \right)^{-1} \tag{9}$$

Using the Woodbury matrix identity, we get:

$$\Sigma_{t+1|t+1}^{(k)} = C_{t+1|t+1}^{(0)} + C_{t+1|t+1}^{(0)} \bar{G}^{(k)T} \left(\bar{W}^{(k)-1} - \bar{G}^{(k)} C_{t+1|t+1}^{(0)} \bar{G}^{(k)T} \right)^{-1} \bar{G}^{(k)} C_{t+1|t+1}^{(0)} \tag{10}$$

The use of this formula can speed up the calculation of $\Sigma_{t+1|t+1}^{(k)}$, because the matrix to invert is usually much smaller than the whole covariance matrix. Standard matrix inversion algorithms require around $2/3n^3$ basic operations, so the computational load is significantly reduced (from almost a million to less than a hundred).

The new Kalman gain is given by:

$$K = \Sigma_{t+1|t+1}^{(k)} G^{(k)T} W^{(k)} = C_{t+1|t+1}^{(0)} G^{(k)T} W^{(k)} + C_{t+1|t+1}^{(0)} \bar{G}^{(k)T} \bar{W}^{(k)} \left(I - \bar{W}^{(k)} \bar{G}^{(k)} C_{t+1|t+1}^{(0)} \bar{G}^{(k)T} \right)^{-1} \bar{G}^{(k)} C_{t+1|t+1}^{(0)} G^{(k)T} W^{(k)} \tag{11}$$

Where we have highlighted what has already been computed in the all-in-view filter.

New subset covariance

As opposed to the optimal filter, the covariance after the update is not given by $\Sigma_{t+1|t+1}^{(k)}$. Instead, it is given by:

$$C_{t+1|t+1}^{(k)} = \left(I - K\right) C_{t+1|t}^{(k)} \left(I - K\right)^T + K W^{(k)-1} K^T \tag{12}$$

SUBOPTIMAL SUBSET SOLUTIONS: SECOND APPROACH

The second approach can be described much more succinctly. It consists in grouping the faults so that we do not need to run as many filters. For example, instead of running a filter for a fault in satellite *i* and another one for a fault in satellite *j*, we run a filter that is fault tolerant to both *i* and *j*. This will result in a weaker solution position, and therefore larger protection levels. This second approach can be considered a suboptimal subset solution approach because each fault is accounted by a suboptimal filter. For this paper, the groups were formed based on the PRN number, which is mostly equivalent to a random grouping with regard to geometry.

DATA AND PROTECTION LEVEL CALCULATION

We used two types of GNSS data: one collected by a static receiver and one collected by a receiver installed in a car.

The GNSS data collected in road conditions is described in [5] and briefly summarized here:

- 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

Specifically, we choose the open sky conditions.

The static GNSS corresponded to the following conditions:

- Receiver: Trimble NetR9
- 6 hours of static data on November 7, 2018 at Stanford
- GPS (L1C-L2W), GLONASS (L1C-L2P) at 1 Hz
- Truth position from IGS station solutions

The error models are and the protection level calculation is also described in [5]. It is a straightforward adaptation of the ARAIM algorithm described in [6] to a Kalman filter solution. The algorithms were run in MATLAB in a PC (Windows 10, 64 bit OS, with Intel® Core™ i7-8700K CPU@3.70GHz 3.70GHz with 16.0 GB RAM).

EXPERIMENTAL RESULTS

Baseline vs suboptimal subset Kalman filter

The baseline results use the optimal subsets as described in Equation (4). The suboptimal Kalman filter subset solution was implemented as described in Equation (11). Figures 1 a) and b) show the subset solutions corresponding to the baseline approach and the suboptimal Kalman filter approach. We can observe a degradation in the accuracy of the subset position solutions, and also in the covariances (not plotted here). Figures 2 through 4 show the protection levels for both the baseline and the suboptimal approach, as well as the ratio between the two (right side plot).

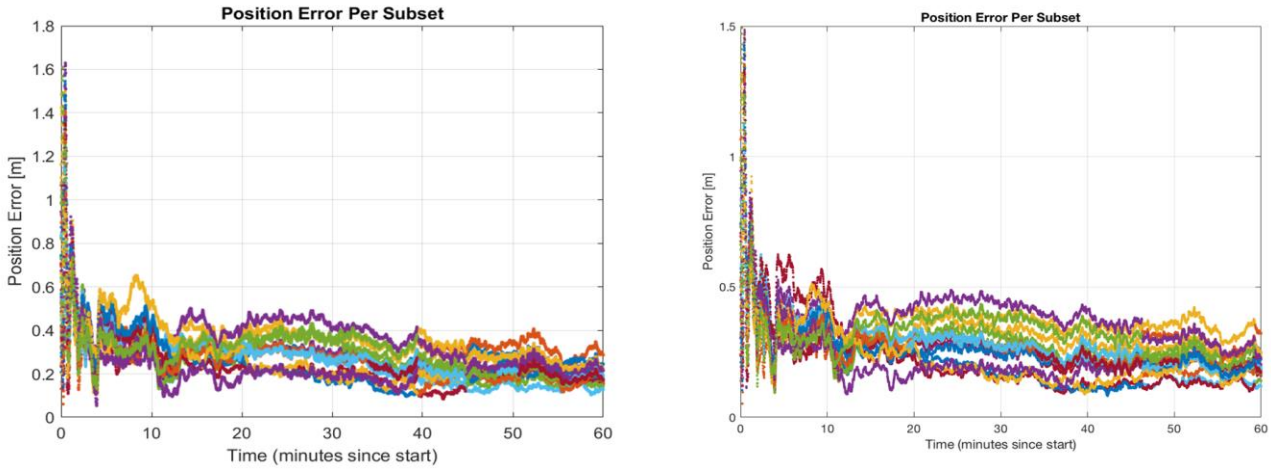

Figure 1 a) and b). Subset position solutions for the baseline results (a), and the suboptimal subset solution (b)

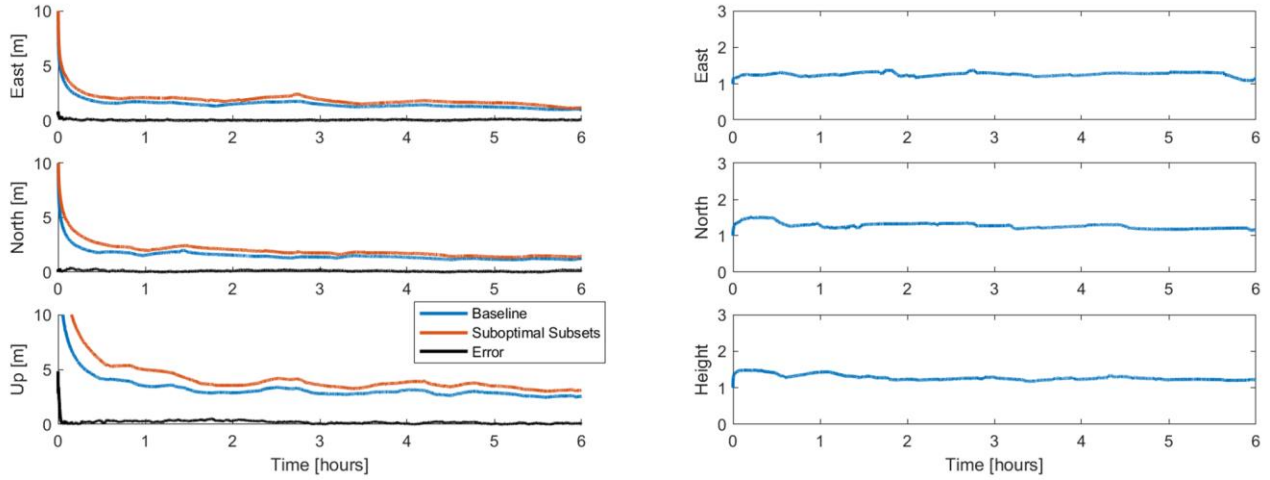

Figure 2. PL for the static scenario with one out (left side plot), and ratio between suboptimal and baseline (right side plot). Run time was 6045 s and 5011 s (for baseline and suboptimal respectively).

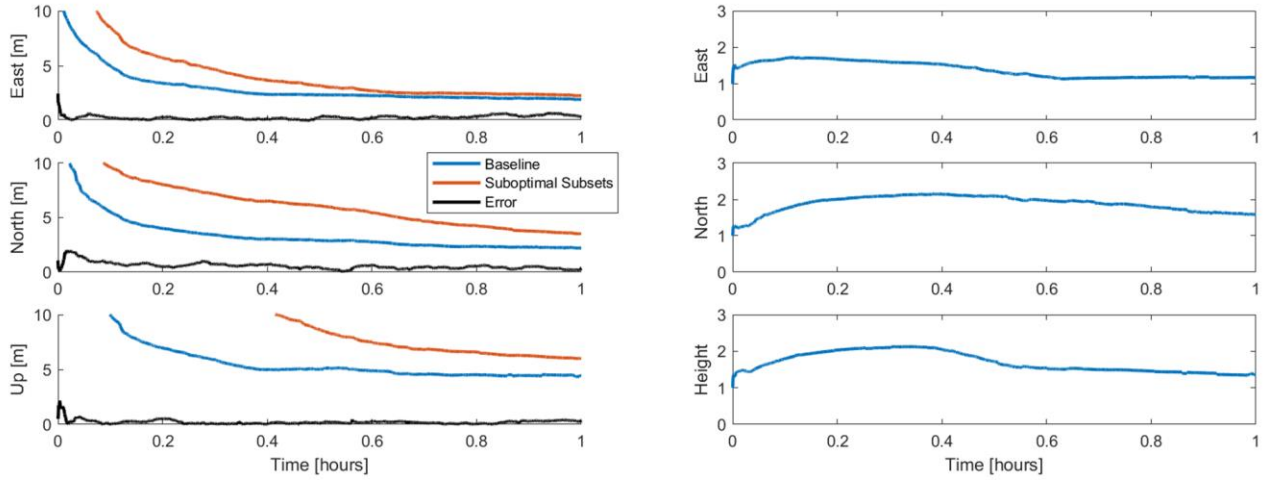

Figure 3. PL for the driving scenario with one out (left side plot), and ratio between suboptimal and baseline (right side plot). Run time was 319 s /291 s (baseline/suboptimal).

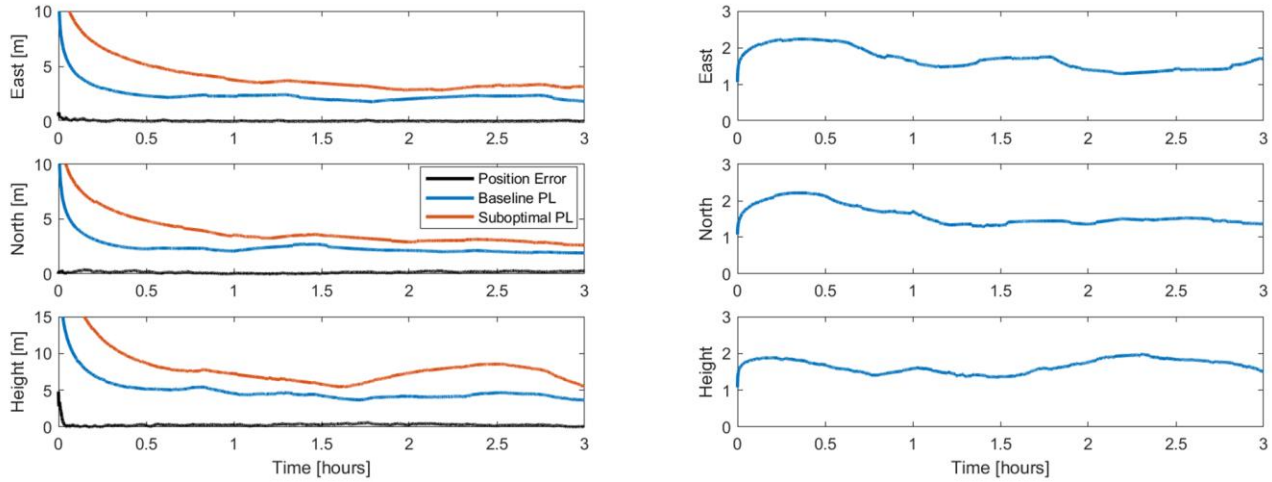

Figure 4. PL for the static scenario with two out (left side plot), and ratio between suboptimal and baseline (right side plot). Run time was 12100 s /6100 s (baseline/suboptimal).

The improvement in execution speed was modest, as we only saw a 10% reduction in execution time for the one out case and a 50% reduction for the two out case. It is possible however that a more significant benefit may be observed in real time code (the prototype we use is implemented in MATLAB, which is very well suited for matrix operations).

Suboptimal subsets: fault grouping

In this section, we evaluate the protection levels resulting from the fault grouping technique. The grouping of the faults was not optimized. Figures 5 through 7 show the resulting protection levels for group sizes of 2, 3, 5 and 10 faults. Up to group sizes of 3, the degradation appears to be acceptable.

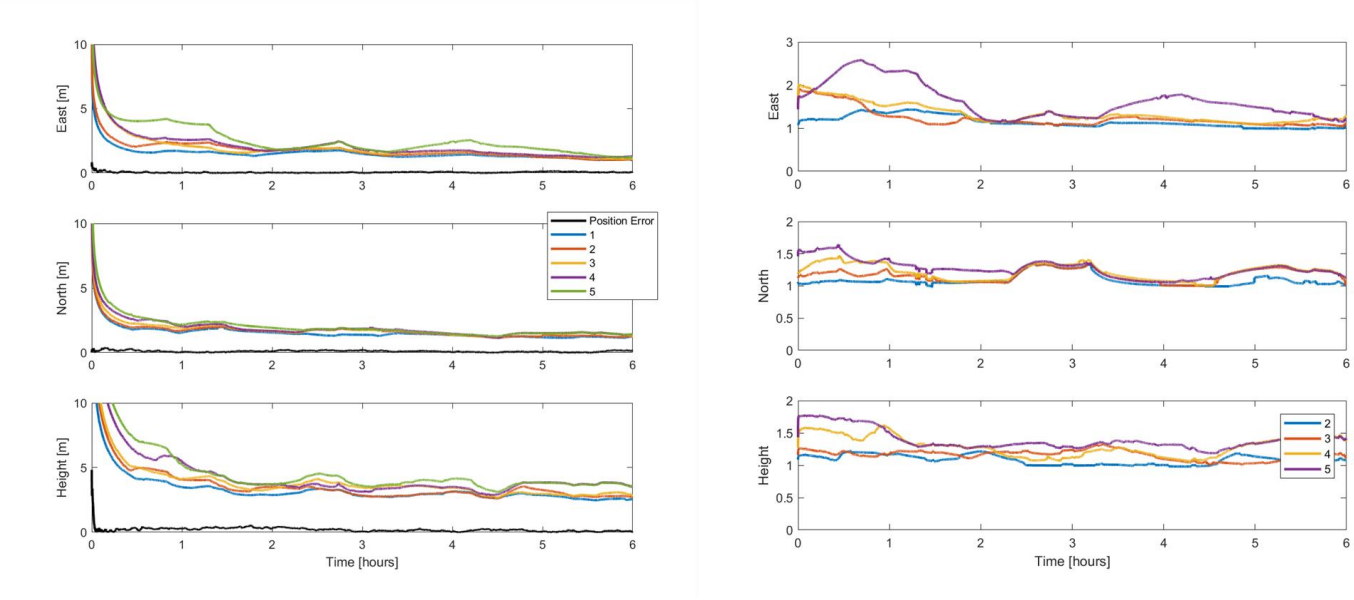

Figure 5. PL for the static scenario with one out (left side plot) for each of the grouping options (1 is the baseline), and ratio between suboptimal and baseline (right side plot). Run time was 6045 s/4324 s/ 3673 s/3352 s/ 3240 s (baseline/2/3/4/5 groupings).

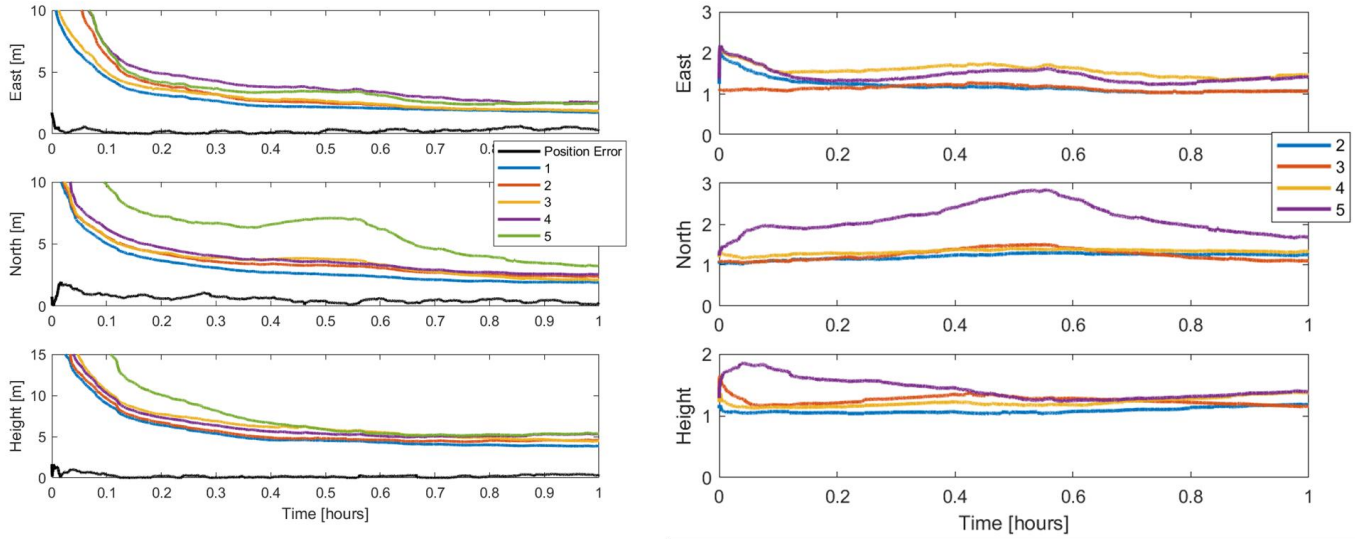

Figure 6. PL for the driving scenario with one out (left side plot) for each of the grouping options (1 is the baseline), and ratio between suboptimal and baseline (right side plot). Run time was 519 s/337 s/ 297 s/273 s/ 265 s (baseline/2/3/4/5 groupings).

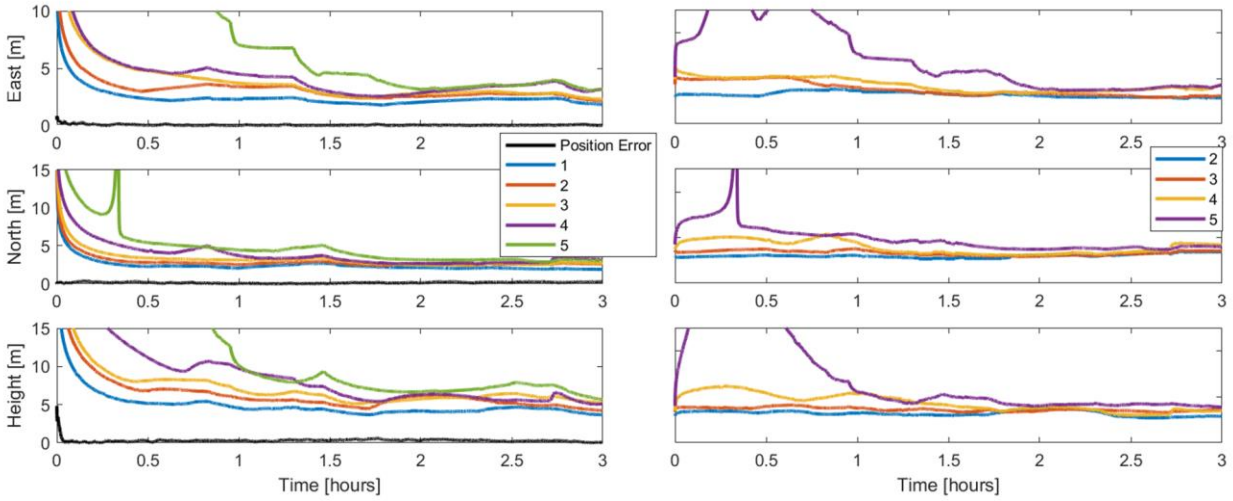

*Figure 7. PL for the static scenario with two out (left side plot) for each of the grouping options (1 is the baseline), and ratio between suboptimal and baseline (right side plot). Run time was 12100 s/**3619 s**/ 2241 s/1787 s/ 1458 s (baseline/2/3/4/5 groupings).*

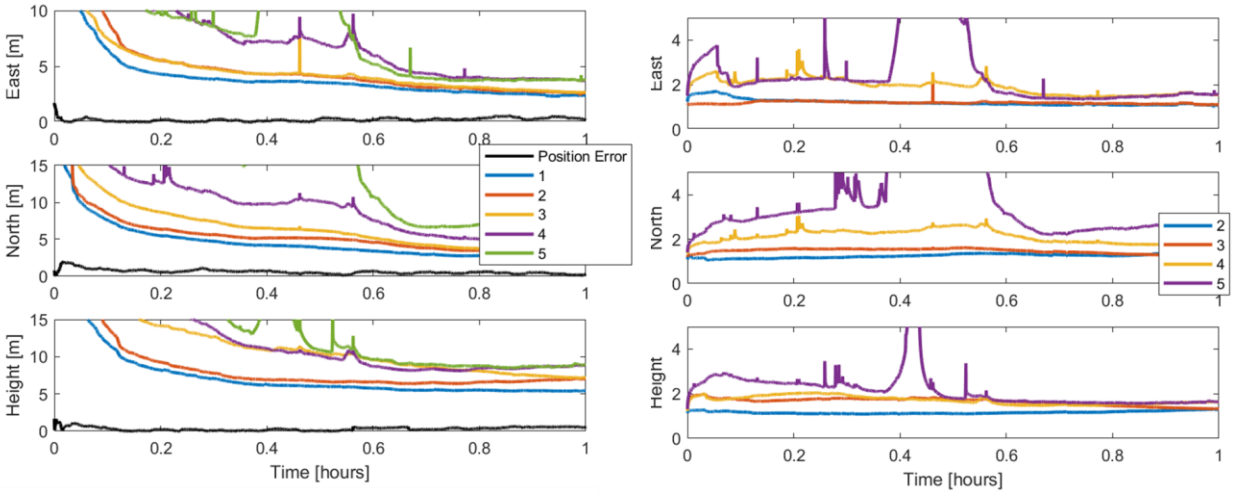

*Figure 7. PL for the driving scenario with two out (left side plot) for each of the grouping options (1 is the baseline), and ratio between suboptimal and baseline (right side plot). Run time was 3800 s/**1280 s**/ 820 s/ 560 s/490 s (baseline/2/3/4/5 groupings).*

As in the first approach, the execution speed improvement in the one out case was modest in our MATLAB implementation, but it is significant in the two out case. For groupings of size two, we observed an improvement of 70% in run time

Forming fault groups with sizes larger than 3 leads to very large protection levels. The protection level degradation due to fault grouping is not negligible in any of the cases, but it may be acceptable for groupings of two faults, especially given the potential reduction in computational load. Table 1 summarizes the results of our simulations for the two out case.

Table 1. Run time improvement for the different techniques and corresponding performance degradation

Algorithm	Decrease in run time compared to baseline	Ratio of protection levels (max and median)	
Baseline solution separation	0 %	1	1
Suboptimal filter	49%	2.23	1.65
Fault grouping (2)	70%	1.54	1.29
Fault grouping (3)	81%	2.07	1.39
Fault grouping (4)	85%	2.44	1.60
Fault grouping (5)	88%	135	2.30

SUMMARY

We have described and investigated two techniques to reduce the computational load of a solution separation algorithm for Kalman filter position solutions. Both techniques consist in using suboptimal filters in the subset filters instead of the optimal filters. We apply these techniques in a solution separation integrity monitoring algorithm in a PPP Kalman filter solution with data collected in road conditions. The results suggest that, although the use of suboptimal filters does increase the protection levels – and therefore degrade performance -, the degradation may be acceptable given the potential computional load savings.

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] D. Calle, E. Carbonell, P. Navarro, P. Roldán, I. Rodríguez, G. Tobías, "Facing the Challenges of PPP: Convergence Time, Integrity and Improved Robustness" *Proceedings of the 31th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2018)*,Miami, Florida, September 2018.

[4] Barrios, J. et al "Update on Australia and New Zealand DFMC SBAS and PPP System Results" *Proceedings of the 31th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2018)*,Miami, Florida, September 2018.

[5] K. Gunning, J. Blanch, T. Walter, L. de Groot, L. Norman, "Design and Evaluation of Integrity Algorithms for PPP in Kinematic Applications" *Proceedings of the 31th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2018)*,Miami, Florida, September 2018.

[6] 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

[7] 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.

[8] 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.

[9] Orejas, Martin, Skalicky, Jakub, "Clustered ARAIM," *Proceedings of the 2016 International Technical Meeting of The Institute of Navigation*, Monterey, California, January 2016, pp. 224-230.

[10] Orejas, Martin, Skalicky, Jakub, Ziegler, Ute, "Implementation and Testing of Clustered ARAIM in a GPS/Galileo Receiver," *Proceedings of the 29th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2016)*, Portland, Oregon, September 2016, pp. 1360-1367.

[11] Tanil, Cagatay, Khanafseh, Samer, Joerger, Mathieu, Pervan, Boris, "Sequential Integrity Monitoring for Kalman Filter Innovations-based Detectors," *Proceedings of the 31st International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2018)*, Miami, Florida, September 2018, pp. 2440-2455.