Airborne Mitigation Of Constellation Wide Faults

Todd Walter and Juan Blanch
Stanford University

ABSTRACT

Advanced Receiver Autonomous Integrity Monitoring (ARAIM) is being investigated as a promising method to provide vertical guidance for aircraft. One of the most significant concerns raised for ARAIM, is the possibility of multiple simultaneous faults being present on one of the constellations. This threat can be mitigated by forming position solutions that exclude each constellation in turn. Unfortunately, if the remaining constellations are not sufficiently strong, the resulting subset can lead to a significant loss of availability. This paper examines some additional airborne consistency checks that can identify some of these feared events.

A simple idea is compare broadcast data sets to prior information in order to identify erroneous ephemeris data. Another idea is to use periods of strong geometry to be able to help get through subsequent periods of weak geometry. Further, by validating broadcast ephemeris messages when constellation cross-checks are available, it may be possible to view the messages as trusted when the cross-checks are no longer available. Such an approach may be able to dramatically improve overall system availability. This paper examines the real constellation fault that occurred on April 1-2, 2014 as well as more difficult to detect constellation fault modes.

INTRODUCTION

ARAIM seeks to incorporate new signals and new constellations for use in horizontal and vertical navigation [1]. However, to use ARAIM for vertical guidance it must meet more stringent integrity requirements than today's RAIM. One significant new threat being evaluated is the possibility of simultaneous faults one a single constellation. Integrity against such threats can be maintained by tracking more than one constellation and comparing position estimates that remove one constellation against position estimates that use all observed satellites [2]. Unfortunately, this approach requires that two strong constellations be available at all times in order to facilitate good performance. Alternatively, three or more moderately strong constellations will also yield high availability.

Other approaches are also possible. It has been suggested that that multiple simultaneous faults are only sufficiently likely through the broadcast of bad ephemeris data. If this supposition is correct, then replacing the broadcast navigation data with data from a trusted source can mitigate this threat without the need for an airborne constellation cross-check.

Alternatively, the airborne user may be able to place confidence in previously broadcast ephemeris messages during times when two strong geometries are available. Then, during a later period, when the geometries are weaker, be able to continue to trust the ephemeris messages and temporarily not require the constellation cross-check.

There are other potential airborne checks that can be added to the currently described algorithm [2]. For example, rather than starting to use new ephemeris data immediately after its receipt, the aircraft could compare the new broadcast data to the prior validated ephemeris. If it is found to be inconsistent, the aircraft could choose either exclude the use of the new data or of the satellite altogether.

Another tactic is to stagger the incorporation of the ephemerides updates. Typically, GPS ephemerides are all updated every two hours. Rather than using all new ephemerides right after the update, the airborne algorithm could choose to use one new one at a time (and use the prior data sets for the other satellites). Once the position fix with the new ephemeris has passed the standard ARAIM consistency check, another new ephemeris can be evaluated. This process continues, one by one, until each ephemeris is updated or rejected. These last two approaches are similar to each other and would prevent large simultaneous step changes in the broadcast data from affecting the position solution. They do not require that a full constellations worth of satellites to be pulled from the position solution for evaluation.

It is also possible to evaluate the consistency of the all-in-view and the individual constellation position solutions over time, to look for slowly evolving threats that may already be present in previously accepted data. Thus, rather than making a snapshot decision as to whether or not the constellations are in agreement, a longer history of comparisons can be evaluated. Evaluating several minutes (or even hours) of data reduces the effect of multipath and other obscuring error sources allowing for tighter thresholds to be used.

The Global Positioning System (GPS) has proven to be extremely reliable over its operational history [3] [4]. However, many of the new core constellations do not yet have an established long-term track record of performance. It is likely that the new constellations initially will be less reliable than GPS. Further, as we will show, the Russian constellation, Global Navigation Satellite System (GLONASS), has experienced more than one system-wide fault where bad data was transmitted [5]. Most recently incorrect ephemeris data was broadcast on April 1-2, 2014 [6] [7]. This event will serve as the basis for much of our investigation.

CONSTELLATION WIDE FAULTS

A constellation wide fault arises when a single event or cause leads to concurrent faults on more than one satellite. This is in contrast to satellite faults that are assumed to be independent from each satellite to any other. Satellite faults require two or more independent unlikely events in order to occur simultaneously. A constellation wide fault requires only a single unlikely event to occur in a way that can affect more than one satellite. A constellation wide fault may affect only two satellites, or any larger number including affecting all of the satellites in the constellation. Further, such faults may be self-consistent or each satellite fault may correspond to very different position and clock errors.

There are several potential causes of constellation faults. The most likely source would be due to incorrect ephemeris data as computed by the ground. One example would be large measurement errors corrupting the satellite clock or position estimates in the ground control system. It may also be possible for operator actions to lead to satellite errors. This could happen by inadvertently uploading incorrect data or by maneuvering satellites without first setting them unhealthy. Another possibility would be a design flaw in the satellites themselves, such as incorrectly handling the week number rollover (when the GPS week number goes from its maximum value of 1023 back to 0). Alternately, it may be possible that a severe solar event could lead to simultaneous upset events on more than one satellite.

It is possible that these latter scenarios may be sufficiently unlikely as to be neglected. However, great care should be taken to quantify their probability before they are summarily dismissed. Nevertheless, this paper will primarily concentrate on mitigating constellation threats that arise from erroneous broadcast navigation data.

MULTIPLE HYPOTHESIS SOLUTION SEPARATION USER ALGORITHM

An example ARAIM user algorithm that has been described in literature is called the Multiple Hypothesis Solution Separation (MHSS) algorithm [2]. We will use MHSS to investigate the performance impacts caused by introducing additional consistency checks. The methods elaborated here can be equally applied to other user algorithm approaches, as the key element is that tests can be constructed that evaluate different fault cases. The specific MHSS algorithm that we use has been developed in cooperation with several other research organizations. Matlab scripts implementing this MHSS algorithm can be downloaded from the Stanford University GPL lab website [8].

The validity of the MHSS algorithm depends on at least one of the subsets containing only unfaulted satellites. Provided such a subset exists, then the nominal covariance for that position estimate should correctly describe a region containing the true position. This subset solution is compared to the all-in-view solution, which may contain faulted satellites (and therefore potentially an incorrect position estimate). If they disagree by more than the internal threshold, the all-in-view position estimate is declared invalid and not used. If they agree to within the threshold, the sum of the threshold and the subset covariance error bound is sufficient and the all-in-view position error will be bounded by the protection levels. The MHSS algorithm also incorporates exclusion. In the event of disagreement, each subset is investigated to see if it is self-consistent. If so, then the satellite that was removed to form that subset is identified as faulty and excluded.

In this paper, two different values are considered for the probability of constellation fault (P_{const}): 10^{-4} and 10^{-8}. The first value corresponds to approximately one hour-long constellation fault per year (or more precisely per 417 days). The second value corresponds to about one

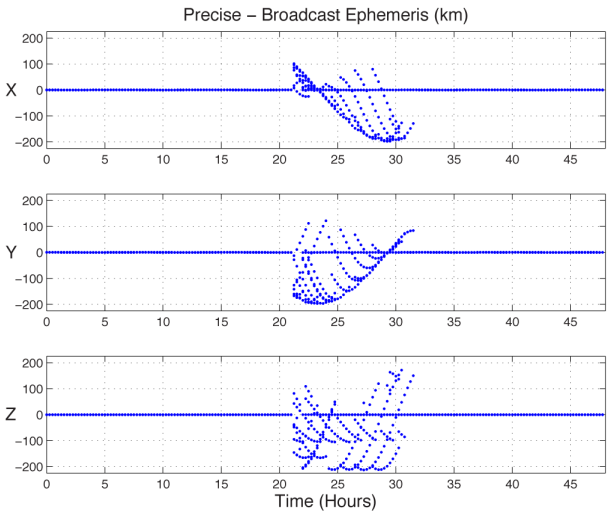

Figure 1. *Broadcast orbital positions versus precise estimates for GLONASS satellites observed at Stanford University on April 1-2,2014*

hour-long constellation fault per 11,400 years. The first value is very conservative compared to the operational history of GPS, which has no known constellation faults during its nearly twenty year operational history (although GPS did experience one such fault before it was declared operational [9]). It seems clear that the correct number should probably be set between these two values, however, proving smaller values is very difficult. A very important distinction between these values is that for 10^{-4}, constellation out subsets must be evaluated. That is, a subset excluding all of the satellites from that constellation must be evaluated. Conversely, if the value is 10^{-8}, this fault mode can be neglected and only single satellite out subsets need to be evaluated. This is because we use a value of 10^{-5} for the independent satellite fault, P_{sat}. This last value corresponds to approximately 2.6 cumulative satellite fault hours per year for a 30 satellite constellation. GPS has averaged well below this number for the last several years [4] [5].

EXAMPLE CONSTELLATION WIDE FAULT

On April 1, 2014 at 21:00 UTC, GLONASS began broadcasting incorrect navigation data on multiple satellites at once [6] [7]. The new ephemeris data had the wrong satellite positions by up to 225 km (see Figure 1). The fault was in the broadcast navigation data and all observed satellites suddenly switched to having large errors upon receipt and application of the new broadcast ephemerides. As we will shortly see, the errors were sometimes self-consistent in that they could pass a RAIM check while still resulting in a position error that was off by tens or hundreds of km. The problem persisted for ten and one half hours. The satellites were gradually corrected until all were again broadcasting valid ephemeris data by 07:30 UTC on April 2, 2014.

For GPS we used the broadcast user range accuracy (URA) value for the integrity sigma (the MHSS algorithm scaling factor, α_{URA_GPS}, was set to 1) and set the accuracy sigma to half that value (α_{URE_GPS} = 0.5). The nominal GLONASS pseudorange errors observed at our Stanford receiver are about twice as large relative to the broadcast URA. We therefore used twice the URA value for the GLONASS integrity sigma ($\alpha_{URA\_GLONASS}$ = 2) and set the accuracy sigma to half that value ($\alpha_{URE\_GLONASS}$ = 1).

Figures 2 and 3 show the horizontal triangle charts for GPS-only and GLONASS-only performance. Triangle charts plot the position error on the x-axis and the protection level (PL) on the y-axis. Hazardously Misleading Information (HMI) occurs if the error is greater than the protection level. When this occurs, pixels

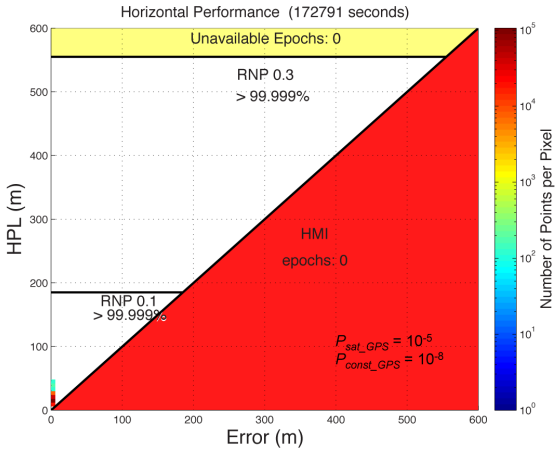

Figure 2. *GPS horizontal performance, April 1-2, 2014*

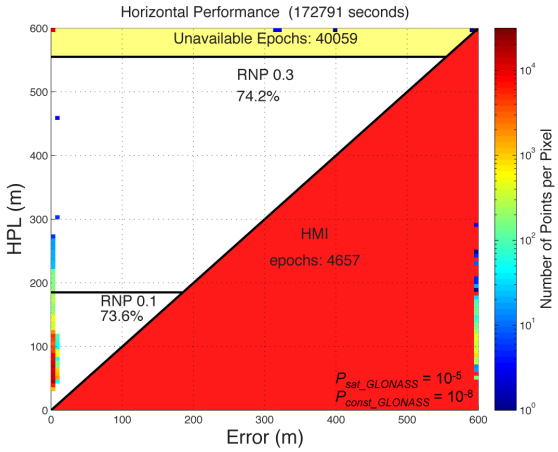

Figure 3. *GLONASS performance, April 1-2, 2014*

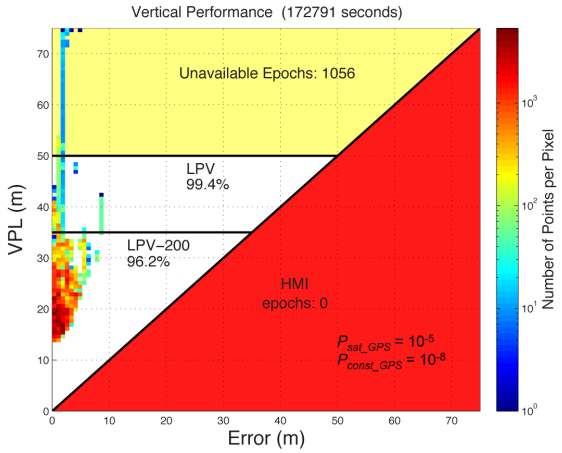

Figure 4. *GPS vertical performance, April 1-2,2014*

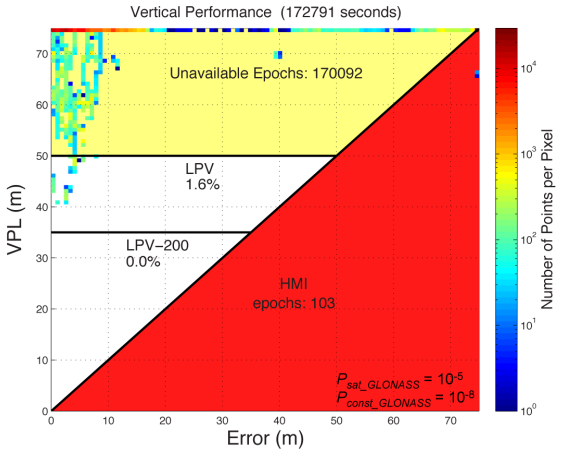

Figure 5. *GLONASS performance, April 1-2,2014*

are plotted below the diagonal line (in the red triangle) and the HMI count indicates for how many seconds integrity was violated. If the PL is too large then the system cannot be used for its intended operation. The number of seconds for which this occurs in listed in the yellow region at the top. Finally, the white triangles show what percent of the time the system was both safe and available for the specific data set and operation.

In each figure, the probability of an independent satellite failure is set to 10^{-5} and the probability of a constellation-wide failure was set to 10^{-8}. This value is not necessarily thought to be valid, but is chosen in order to only evaluate single satellite out geometries. Also, it is chosen because single constellation solutions cannot be formed if P_{const} is set above 10^{-7}. GPS, in particular, gives very good availability against RNP 0.1 and has no integrity faults. GLONASS, in contrast, has worse availability and does experience integrity faults. The horizontal error exceeded 50 km even while the ARAIM horizontal protection level (under the incorrectly assumed probabilities) was below 300 m. Clearly, assuming a constellation wide-fault probability of 10^{-8} for GLONASS is an incorrect and unsafe choice for these specific days.

It is even more instructive to look at the vertical performance. Here the margin between nominal accuracy and the alert limit is much more narrow. Figure 4 shows the vertical performance for GPS-only using the same data. Availability is high, despite there being only one constellation. This high availability is due to the very strong GPS geometry and the negligible probability of constellation wide fault. Figure 5 shows the same days for GLONASS. GLONASS has fewer overall satellites, worse accuracy, and worse geometry, leading to lower availability. The constellation-wide fault, has a smaller impact on the vertical position, but still creates vertical errors up to nearly 12 km. Also, there are far fewer times where the vertical position error exceeds the vertical position error.

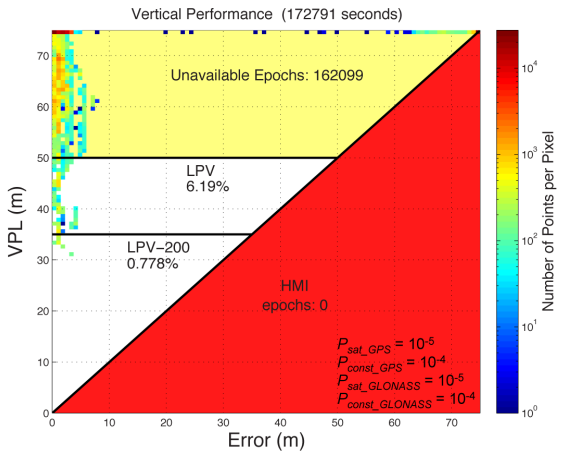

Figure 6. *GPS & GLONASS vertical performance assuming constellation faults, April 1-2,2014*

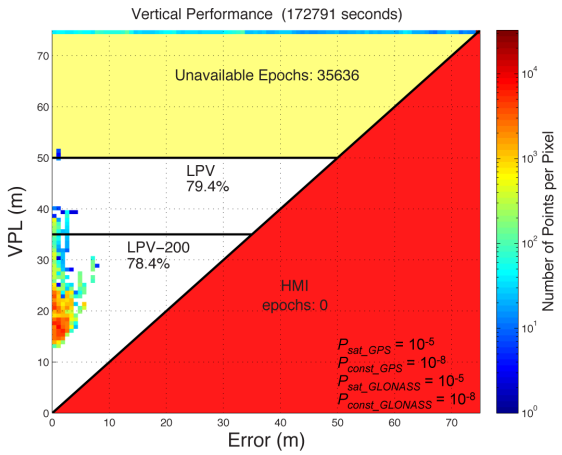

Figure 7. *GPS & GLONASS vertical performance assuming no constellation faults, April 1-2,2014*

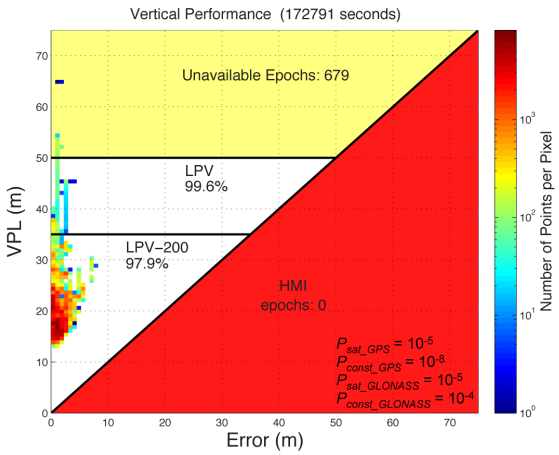

Figure 8. *GPS & GLONASS vertical performance assuming constellation faults on GLONASS only*

The constellation cross-check is advised exactly for scenarios such as this where it is sufficiently unlikely that two different constellations would share a common fault mode. Figure 6 looks at the combined use of GPS and GLONASS where P_{sat} has been set to 10^{-5} and P_{const} has been set to 10^{-4} for each constellation. As expected, the MHSS algorithm correctly identifies that a problem exists with one of the constellations, but given that there are only two, and they are equally trusted, it cannot determine which one is faulted. Further the MHSS algorithm cannot operate with only constellation one even if it could isolate the faulty one (given $P_{const} = 10^{-4}$ for the remaining constellation). Figure 6 shows much reduced availability compared to the GPS only case in Figure 4. This is because one of the subsets formed only has GLONASS satellites). Availability cannot be better than the worst performing constellation in this case. The comparison to the GPS-only case is unfair, as Figure 4 does not account for the possibility of the constellation wide fault. Nevertheless it does provide motivation to identify and remove such faults to whatever extent possible.

Figure 7 looks at another scenario where P_{sat} has been set to 10^{-5} and P_{const} has been set to 10^{-8} for each constellation. This is not a reliably safe approach, but for this fault case, the difference between the single constellation position solutions is too large to ignore during this GLONASS fault event. Notice that the VPL is much lower due to not having to evaluate single constellation subset geometries. The overall availability is much improved compared to the prior case. However, there are still nearly ten hours of unavailability as the MHSS algorithm cannot isolate the fault. This inability to isolate is due to only testing single satellite out subsets. The ability to remove the full faulty constellation while trusting the remaining unfaulted one would lead to the highest availability.

Figure 8 shows a more ideal case where we have set P_{sat} to 10^{-5} and P_{const} has been set to 10^{-8} for GPS and to 10^{-4} for GLONASS. These settings make use of posteriori knowledge that GLONASS fails on these days and that GPS does not. However, it serves as a good target for performance. Notice the high availability. For 37.5 hours GPS and GLONASS work well together providing excellent geometry. And, for 10.5 hours GPS works well on its own while GLONASS experiences its constellation wide fault. There are only about 11 minutes over the two days where LPV was not available under this set of assumptions. Notice that compared to Figure 4, the overall VPLs are lower and availability has improved.

DELAYED INCORPORATION OF EPHEMERIDES

The GLONASS fault appeared to be a sudden constellation wide fault, because the broadcast ephemeris information for all of its satellites updated and changed at the same time. However, if the ephemeris information were staggered, such that only one satellite were updated at any given epoch, then we could create a situation where there was only one faulted satellite being considered at each time. At start up, there would be no history and no established trust in either of the constellations. However, if the receiver can successfully initiate by performing the constellation cross-check when each constellation has adequate geometry, then perhaps it can continue to operate through faults and through weaker geometries.

For the moment, let us assume that previously checked ephemerides can be used at later epochs (we will examine slowly growing faults in a later section). In this case, at start-up, the receiver operates with P_{sat} and P_{const} set to 10^{-4} for each constellation. Once it has successfully determined that the two constellations are consistent and in agreement, it can set P_{const} set to 10^{-8} for each constellation.

At a later time new ephemeris data is received. The health status is immediately checked and any satellite indicating that it should not be used is discarded. If more than one satellite has received an update that epoch, then all but one are delayed until later epochs. One is chosen for evaluation. If it is accepted, the satellite continues to be used as before, but now with the new broadcast ephemeris data. If it is rejected, then a choice could be made either to continue to use the old ephemeris data or instead to reject the satellite altogether. After this choice

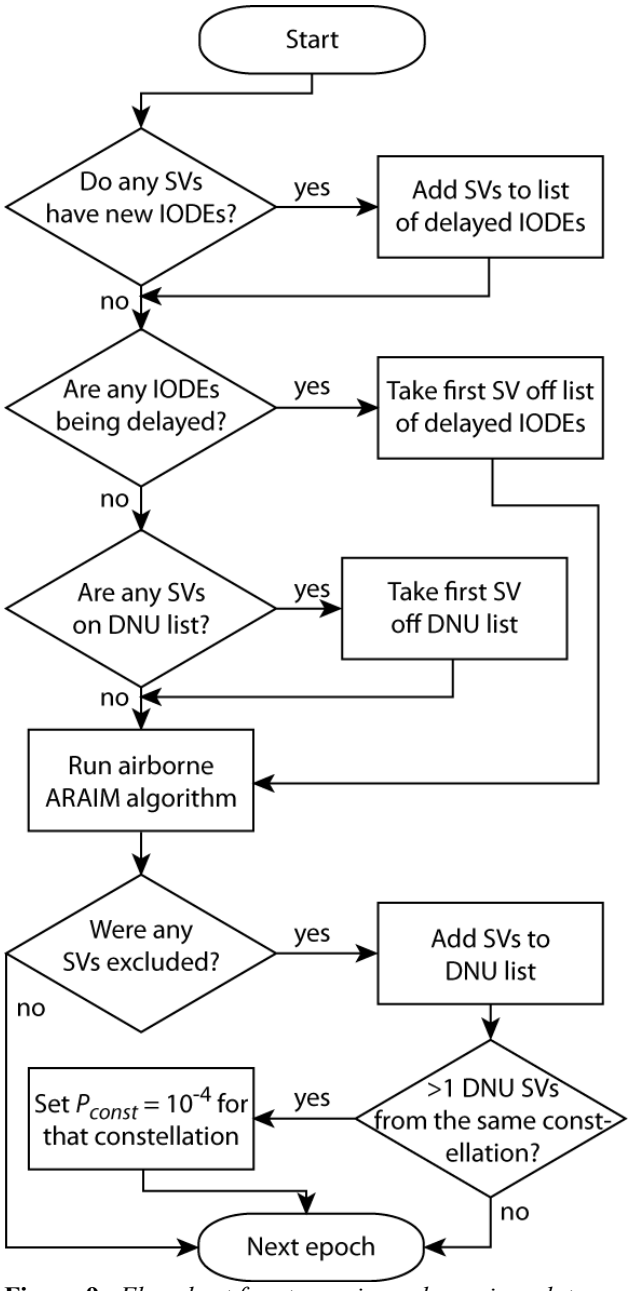

Figure 9. *Flowchart for staggering ephemeris updates*

is made, another epoch runs either with the old data or without the satellite. The next satellite with an update is tested the next epoch. This process is repeated until all of the new data have been evaluated. Most commonly, all new data will be accepted and the new broadcast ephemerides will be delayed by a few seconds at most.

Figure 9 is a flowchart depicting the algorithm that was implemented to stagger the ephemeris updates. Not shown is an initial check for satellite health. If the current satellite ephemeris indicates that the satellite is unhealthy, it should not be used until the health bit is again set healthy in the broadcast navigation data. Otherwise the satellite is considered usable.

First, it is checked to see if the ephemeris information has changed (the flowchart uses the shorthand notation Issue Of Data: Ephemeris, or IODE, but really it should be checking to see if any of the data content has changed). Any of the satellites with changed ephemeris data are added to a list of satellites that are to delay using their new information. Next, if there are any satellites on this list, the first one is taken off and it will use its new ephemeris data. Any other satellites on the list will continue to use their prior data. The next decision point in the flowchart has to do with reintroducing previously excluded satellites and we will describe this process later.

The set of satellites with either all previously tested ephemerides, or at most, one untested new ephemeris is put into the MHSS ARAIM algorithm. If this new ephemeris contains bad information, the MHSS algorithm can likely exclude it as it is testing all one satellite out cases. Most likely the new data is valid and the satellite will not be excluded. In the event that it is excluded, it is added to a do not use (DNU) list and excluded from further epochs. It may be possible to return to using the prior ephemeris for an extended time. However, we did not further evaluate such an optimization for this paper.

Once a satellite has been excluded, there also needs to be consideration for how it can eventually be trusted for use again. If the satellite were truly faulted, then ideally it would remain excluded until the fault is removed. After the fault is removed, the aircraft would want to include it again in its position estimation. If the exclusion is a false alert, then ideally it should be reincorporated as soon as practical. Unfortunately the aircraft does not know which of these two situations apply, nor would it know when a true fault is removed. Therefore, it must periodically re-evaluate the satellite and decide if it still thinks a fault is present.

This reintroduction scheme should involve some memory, that is, once excluded it should probably remain excluded for some period of time. However, too long a time could lead to lower continuity/availability, especially if false alerts are sufficiently likely. Too short a time could lead to integrity problems, as the likelihood of fault given a prior exclusion may be much higher than the MHSS algorithm assumes. It might also be desirable to increase P_{sat} for the excluded satellite, at least until it again passes a consistency check. Because we have a short data set and because we wanted to stress test our new algorithm, we imposed neither a time holdout nor a P_{sat} increase.

For an operational staggering algorithm we would advise including both.

We now return to the decision point above running the ARAIM algorithm. If none of the satellites are having new ephemeris data delayed, then one excluded satellite can be taken off of the DNU list and be tested. Again this should conform to the expectation that only one faulted satellite is being presented to the ARAIM algorithm at a time. If the satellite under test is found to be consistent with the other satellites, it can again be used. Otherwise, if it is again excluded, it is added to the bottom of the DNU list.

In the case of a single faulty satellite, this means that this satellite will always be evaluated, as it is the only satellite on the DNU list and is therefore taken off the list every epoch (again we emphasize that this would be a bad practice in real operation). If there are two faulted satellites, they alternate every epoch as to which one is tested and which one is kept out of the ARAIM algorithm.

In the example of the April 1st fault, at 21:00 all eight GLONASS satellites in view received updated ephemeris information. That epoch, one GLONASS satellite started using this new information and the other seven continued to use the prior broadcast data. The first satellite was excluded and added to the DNU list. The next epoch another satellite started using the new data and was similarly excluded. After six epochs, six GLONASS satellites were labeled DNU and two were yet to be tested. When testing the last two, the ARAIM algorithm could not properly distinguish between faulted and unfaulted GLONASS satellites. This situation occurs because each constellation has its own clock state (i.e. we are estimating five unknowns instead of four unknowns as we would for a single constellation). Thus, both subset geometries that have only one GLONASS satellite are equally valid. A constellation with a single satellite, only uses its range measurement to determine the constellation time offset. It cannot create any inconsistency compared to the other constellation's measurements no mater how large its error is. Therefore, one of the two satellites will somewhat be arbitrarily be marked faulted and the final satellite is never found to be inconsistent.

In running this version of staggering the updates, we found that at each epoch after the first seven, two GLONASS satellites would be tested, one would be excluded and added to the bottom of the DNU list and the next epoch repeats this process testing different pairs in turn. Eventually, after several hours, our new algorithm presented a pair and MHSS did not exclude either of them. Evidently, they were consistent enough with each other, that the GLONASS clock state could absorb the error. The next epoch then presented three faulty GLONASS satellites to the MHSS algorithm. At this point, the fundamental idea of staggering updates and submitting no more than one (or two if they are the only two satellites in the constellation) faulty satellite to the MHSS algorithm has been violated. Shortly thereafter, the position errors and PLs grew very large and the MHSS algorithm was unable to eliminate enough of the faulty satellites as it was only evaluating single satellite out sets. Holding out the satellites for longer periods of time could have easily solved this issue, but it points to a larger concern that if the MHSS algorithm can't always successfully exclude a satellite with new and invalid information, then the basic approach of staggering is flawed.

Further, it does not make sense to assume that P_{const} is 10^{-8} when there are more than one excluded satellites from the same constellation. We therefore added the final decision point shown in the flowchart: if more than one satellite from the same constellation is added to the DNU list then set P_{const} to 10^{-4}. While it is debatable whether even this latter value is valid given our knowledge, the important aspect is that now the MHSS algorithm will evaluate removing all satellites from that constellation. This updated algorithm proceeds exactly as before because the faults are still initially presented to it one at a time due to staggering the ephemeris updates. After the second satellite is updated and excluded P_{const} is set to 10^{-4} for GLONASS. The next satellites are excluded one by one. Eventually all GLONASS satellites are rotating on and off of the DNU list. Now, when the algorithm finds a pair that is initially consistent, it rapidly recovers because it can exclude all GLONASS when they are no longer consistent.

There also should be logic to again set P_{const} to 10^{-8} after the fault is over. Although, it is also correct to argue that 10^{-8} is an unreasonably small number given that such an event has already been observed. At the moment, we are not arguing the validity of these values, but are rather using them to either enable or disable constellation out subset evaluation in the MHSS algorithm. We also need logic to initialize probabilities after a cold start. We do not wish to create an algorithm whose performance is dramatically different depending upon when the receiver is first turned on. Our goal would be to have the receiver perform identically on April 3rd (after the fault is long gone), whether it was first started on April 1st (before the fault), April 2nd (during the fault), or April 3rd. Thus, we don't want to build extended memory into our algorithm; it should quickly revert to a normal mode under nominal conditions with good geometries.

Figure 9 does not show the recovery logic after setting P_{const} to 10^{-4}. Also not shown is the initialization state. Ideally, we would return P_{const} to 10^{-8} after a period of time where we have no satellites on the DNU list and we have very strong geometries and very good agreement across all satellites (e.g. small solution separations or χ^2 values). Determining sufficiently tight validation requirements requires a lengthy analysis that is beyond the scope of this paper. Instead we used an ad hoc approach that will surely need to be refined. We required that there be no satellites on the DNU for 300 consecutive seconds and that during that entire time all LPV-200 requirements be met (VPL $\leq$ 35 m, HPL $\leq$ 40 m, EMT $\leq$ 15 m, and $\sigma_v \leq 1.88$ m [1]). A full safety analysis would likely result in even stricter requirements in order to set P_{const} to 10^{-8}. Further, the algorithm should be initialized with P_{const} set to 10^{-4} for all constellations. The values would only go down to 10^{-8} after the adequate validation requirements are met.

Unfortunately, as can be seen in Figure 6, when P_{const} is set to 10^{-4} for both constellations in our sample data set, and assuming $\alpha_{URA\_GLONASS} = 2$, we have no times when all of the LPV-200 requirements are met. In the future, when the broadcast URAs will be smaller, and the constellations have stronger geometries, it will hopefully be easier to meet these or the appropriately strict requirements. For this data set, we initialized both constellations with $P_{const} = 10^{-8}$ and used our ad hoc requirements (which have good availability for $P_{const\_GPS} = 10^{-8}$ and $P_{const\_GLONASS} = 10^{-4}$ as seen in Figure 8).

Figure 10 shows the performance of the described algorithm against our example threat. As can be seen, it does better than any of the other evaluated scenarios. It always excluded the inconsistent satellites when they were faulted and it made optimal use of GLONASS when it was unfaulted. Effectively, $P_{const\_GLONASS}$ was 10^{-8} before and after the fault period and was set to 10^{-4} a few seconds after the fault started, and remained there until approximately five minutes after the fault ended. Admittedly, the algorithm is somewhat tuned for this specific fault, so it is not surprising that it performs well on this data set. The important question is how it would perform on future faults and what can be done to make it more robust and more rigorous?

SMALL OR SLOWLY GROWING FAULTS

The above approach is not sufficient to mitigate other threat scenarios. It is well suited for the observed fault case where a large step error is present in one or more

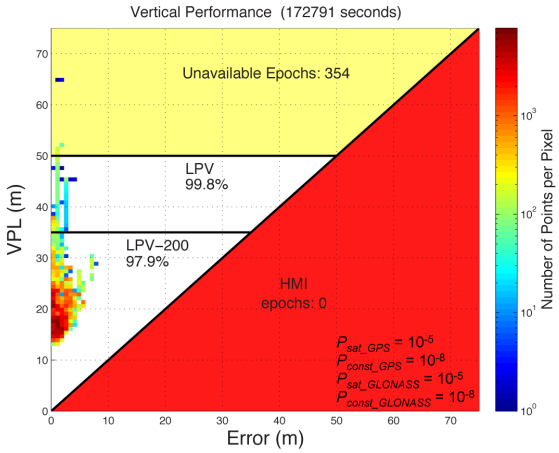

Figure 10. *Vertical performance for the full algorithm*

broadcast navigation data sets, but it may not be sufficient against smaller step errors. Nor does it necessarily protect against slowly growing ramp errors. For such threats, a better approach may be to monitor the difference between the two single constellation-out position solutions compared to the all-in-view solution over time. Both the magnitude of the differences and the rates of change should be evaluated. A long evaluation period with good results may be able to lead to successful coasting through a period with weak geometry.

Unfortunately, the data that we collected for the prior analysis was from a static location that has significantly more multipath than would be expected in an airborne environment. Further, the accuracy of the GPS and GLONASS signals are yet as good as we would like. They are expected to improve by the time both L1 and L5 signals are available for all of each constellation. Instead, we now look at the expected accuracy assuming airborne multipath (AAD-B [10]) URE values of 0.75 m and vertical tropospheric uncertainties of 5 cm 1-σ.

Figure 11 shows the expected vertical accuracies for both GPS and Galileo at Stanford University using recent almanacs for both constellations. For reference the LPV-200 accuracy threshold of 1.88 m is shown [1]. We assume that similar levels of accuracy may be required from each constellation in order to successfully cross-validate their individual position errors. More importantly the figure shows the quickly varying nature of the expected accuracy as the satellites rise and set. Notice that the spikes above the reference line are fairly short lived, persisting for no more than 30 minutes. Ideally the data below the line could then be used to verify that the existing ephemeris data is likely valid for 10 to 30 minutes.

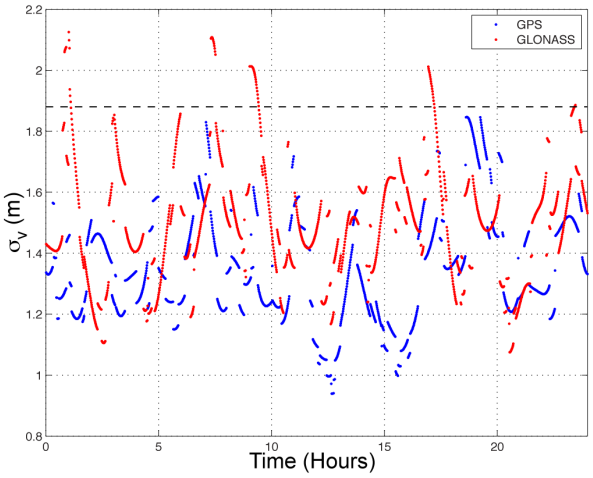

Figure 11. Expected future *GPS & GLONASS airborne vertical accuracy given URE values of 0.75 m for each.*

Figures 12 and 13 show the relative projection error when using the prior broadcast ephemeris data set compared to using the latest one. As can be seen the GPS error difference is generally very small and often unchanging over the next two hours. GLONASS in contrast has a larger initial difference between ephemeris data sets and this error grows noticeably after about ten minutes. The GPS data format is better suited to long-term application and has higher accuracy.

Obviously this test concept is still in the very early stages. Better modeling of the airborne accuracy and the temporal correlation of the error sources is required to properly determine the magnitude of position errors that could be detected and acceptable rates of growth for these errors. One implementation that will be shortly tested is to have the MHSS algorithm evaluate the constellation out subsets even when the P_{const} is set to 10^{-8}.

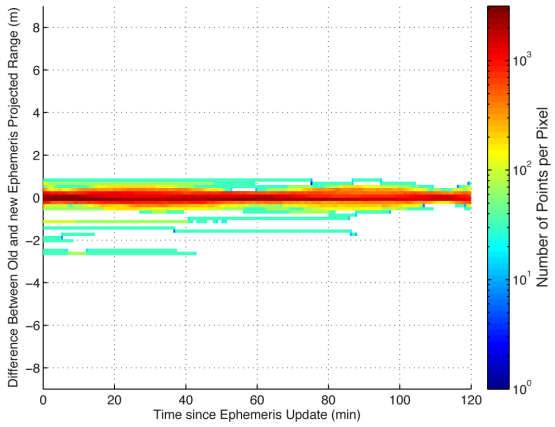

Figure 12. *Prior GPS ephemeris projection error compared to current ephemeris data*

DISCUSSION

The idea to implement additional airborne checks to augment the baseline MHSS algorithm is still in the very early stages. The notion of staggering ephemeris updates adds some value but is not a complete solution. It could be further augmented by monitoring the magnitude and rate of growth of the one constellation out subsets compared to the all-in-view solution. This latter test should be effective against slowly growing errors. However, acceptance of this approach likely requires agreeing to limits on what behavior constellation faults may exhibit. This is not a new idea. All of the currently proposed solutions require some restriction on the constellation fault threat model.

The baseline MHSS algorithm operates on the principle that the constellations are independent of each other. It requires that a common fault cause will not affect more than one constellation at the same time.

Another approach being investigated, online ARAIM, is replacing the broadcast navigation data with trusted information that has been generated by certifiably safe source [11]. Online ARAIM operates on the principle that the only sufficiently likely source of constellation error is due to erroneous data broadcast.

As the approaches in this paper are further investigated they may require similar restrictions on the threat space in order to be accepted as safe. In all cases the limits of the constellation wide threat space require further discussion and ultimately international acceptance. Similar threat models for augmentation systems required years of study and deliberation.

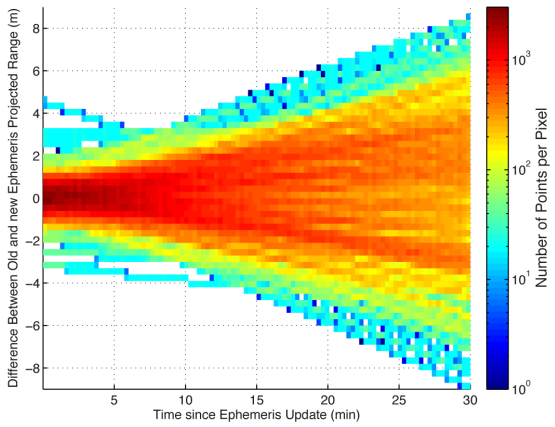

Figure 13. *Prior GLONASS ephemeris projection error compared to current ephemeris data*

CONCLUSIONS

The April 1-2, 2014 GLONASS fault case has been evaluated for its impact on GLONASS only and GPS & GLONASS performance. It was found that this threat included a feared type of event: a consistent set of errors affecting all satellites in view from a single constellation. It was observed that this fault was too large for any reasonably implemented ARAIM algorithm to miss when combining GPS and GLONASS. However, it is necessary to be concerned about potentially smaller, and more difficult to detect, fault modes.

We have coded an ephemeris delay algorithm that prevents sudden changes in multiple broadcast ephemerides from confusing the MHSS algorithm. Without this change, the algorithm is unable to correctly identify and isolate the faulty satellites. With this change, the ephemeris updates are presented to the MHSS algorithm one by one and each newly faulted satellite is readily identified and excluded. We found that re-examination of the excluded satellite is a difficult, but important process in order to again be able to use these satellites after the fault is over. The process of reintroduction requires much more extensive and thorough investigation in order to ensure safe continued operation.

The process of initializing the system also requires greater study. This new algorithm introduces memory into the process. When there is no prior data due to a cold start of the system, cautious assumptions must be made about the state of the satellites. To be safe, one could assume that the satellites all start in a faulted state until proven self-consistent. However this may lead to very low availability. We likely need to find a less conservative, but still safe approach to this problem.

We have also begun to investigate other airborne checks to broaden the range of constellation wide threats that can be detected with our proposed approach. However, there will likely be some hypothetical very difficult to detect constellation wide faults that cannot be mitigated. We will have to investigate if these are sufficiently likely as to warrant concern or if they can be safely neglected. This holds true not just for the approaches investigated in this paper, but for all proposed solutions to addressing the constellation wide threat.

Finally, we also need to investigate the trade between level of onboard complexity and expected benefit. We have specifically investigated the case where we have one strong constellation and one weaker constellation. This is a near worst-case scenario. When we have two very strong constellations, or three moderately strong constellations, the existing MHSS constellation cross check is both effective and has high availability for LPV-200. Therefore the suggested checks in this paper may not be required, depending on what set of constellations are able to be used for aviation. Nevertheless we recommend continued investigation into staggering the ephemeris updates as it may provide an added layer of protection regardless of whatever other algorithms are in place or the future availability of constellations.

ACKNOWLEDGMENTS

The authors would like to gratefully acknowledge the FAA Satellite Product Team for supporting this work under Cooperative Agreement 2012-G-003. However, the opinions expressed in this paper are the authors' and this paper does not represent a government position on the future development of ARAIM.

REFERENCES

[1] Blanch, J., Walter, T., Enge, P., Wallner, S., Fernandez, F. A., Dellago, R., Ioannides, R., Hernandez, I. F., Belabbas, B., Spletter, A., and Rippl, M., "Critical Elements for a Multi-Constellation Advanced RAIM", *NAVIGATION, Journal of The Institute of Navigation*, Vol. 60, No. 1, Spring 2013, pp. 53-69.

[2] Blanch, J., Walter, T., Enge, P., Lee, Y., Pervan, B., Rippl, M., Spletter, A., "Advanced RAIM user Algorithm Description: Integrity Support Message Processing, Fault Detection, Exclusion, and Protection Level Calculation," *Proceedings of the 25th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2012)*, Nashville, TN, September 2012.

[3] FAA SPS PAN Reports available at http://www.nstb.tc.faa.gov/DisplayArchive.htm

[4] Heng, L., Gao, G. X., Walter, T., Enge, P., "GPS Signal-in-Space Anomalies in the Last Decade: Data Mining of 400,000,000 GPS Navigation Messages," *Proceedings of the 23rd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2010*), Portland, OR, September 2010, pp. 3115-3122.

[5] Heng, L., "Safe Satellite Navigation with Multiple Constellations: Global Monitoring of GPS and

GLONASS Signal-in-Space Anomalies," Ph.D. Dissertation, Stanford University, December 2012. (available at: http://waas.stanford.edu/papers/Thesis /LHengThesisFinalSignedSecured.pdf

[6] Beutler, G. Dach, R., Montenbruck, O., Hugentobler, U., Weber G., and Brockmann, E., "GLONASS and Multi-GNSS in the IGS: Lessons learned from GLONASS Service Disruptions," presented at: *13th Meeting of the National Space-Based Positioning, Navigation, and Timing (PNT) Advisory Board*, June 2014. (available at: http://www.gps.gov/governance/ advisory/meetings/2014-06/beutler1.pdf).

[7] van Diggelen, F., "How GLONASS failed for 11 hours, and multi-GNSS receivers survived" presented at *Stanford University PNT Challenges and Opportunities Symposium*, October, 2014. (available at: http://scpnt.stanford.edu/pnt/PNT14/2014_Presentation_Files/7.van_Diggelen-GLONASS-multi_GNSS.pdf).

[8] http://waas.stanford.edu/staff/maast/maast.html

[9] Shank, C. M. and Lavrakas, J., "GPS Integrity: An MCS Perspective," *Proceedings of the 6th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1993)*, Salt Lake City, UT, September 1993, pp. 465-474.

[10] SBAS Minimum Operational Performance Specification (MOPS), RTCA document DO-229D, December, 2006.

[11] Blanch, J., Walter, T., Enge, P., Pervan, B., Joerger, M., Khanafseh, S., Burns, J., Alexander, K., Boyero, J. P., Lee, Y., Kropp, V., Milner, C., Macabiau, C., Suard, N., Berz, G., Rippl, M., "Architectures for Advanced RAIM: Offline and Online," P*roceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014)*, Tampa, Florida, September 2014, pp. 787-804.