

The Stanford - ESA Integrity Diagram:

A new tool for the user domain

SBAS integrity assessment

M. Tossaint(1), J. Samson(1), F. Toran(1) and J. Ventura-Traveset (1),

M. Hernandez-Pajares(2), J.M. Juan(2), J. Sanz(2) and P. Ramos-Bosch(2)

(1) European Space Agency, ESA

(2) Research Group of Astronomy and Geomatics,

Universitat Politècnica de Catalunya (gAGE/UPC), Barcelona, Spain.

ABSTRACT

A new diagram for SBAS integrity validation is presented in this paper. The proposed concept is a modification of the well known Stanford plot, where a 2D histogram shows the relationship of position errors against protection levels for a set of measurements using an all in view satellite selection. The method consists on a new representation (known here as the Stanford-ESA Integrity Diagram) where, at each sample time and given location, the solutions from all possible combinations of satellites from 4 to all -in-view are displayed, instead of only the all-in-view one. In case of MIs, the geometries associated to each epoch are labeled with different symbols and colors. It allows easy identification of the different clusters and assessment of the time correlation of the events. Real measurement results are presented here showing that EGNOS integrity margins remain safe under this very exigent criterion, which is certainly a very positive result.

INTRODUCTION

The Satellite Based Augmentation Systems (SBAS) broadcast through GEO satellites GPS-like navigation signals containing differential corrections and integrity data to enhance the GNSS positioning and making it suitable for safety critical applications such as civil aviation.

Integrity is the system's ability to provide warnings to the user when the system is not available for a specific operation. The SBAS systems provides the users with integrity information to compute the Protection Levels (Horizontal and Vertical protection levels HPL and VPL), which represent an upper bound of the error.

For each operational mode, Alert Limits (AL) against which the user has to compare its Protection Levels are defined (ICAO's GNSS SARPS) [1], and the system is declared Unavailable when the Protection Level is greater than the Alert Limit. If the system is available and the Position Error is not bounded by the Protection Level, thence the event is considered as a Misleading Information (MI), since the Protection Level is always supposed to be an upper bound of the Position Error. Moreover, the event is declared as Hazardously Misleading Information (HMI) if the Position Error exceeds the AL (see figure 1).

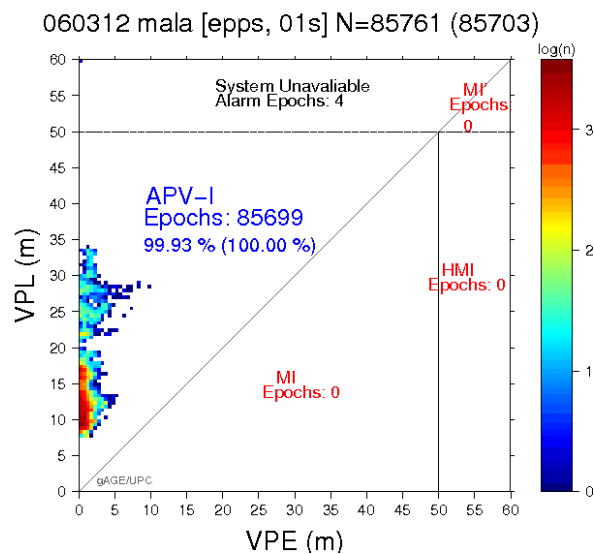


Figure 1: Example of Stanford Plot for the Vertical component. The horizontal axis is the Vertical Position Error (VPE) and the vertical axis the Vertical Protection Level (VPL). The Alert Limit is also shown in the plot as horizontal and vertical lines at 50m. Each bin indicates (in a logarithmic color scale) the number of occurrences of a specific (VPE,VPL) pair.

The Stanford Plot has become a reference representation technique in the SBAS domain, especially to have a quick and clear view of system performances, highlighting its capability to clearly show the integrity margins offered by SBAS. This Diagram uses an all-in-view approach (i.e. all GPS satellites in view with valid differential corrections available) for computing the error/protection level pair (XPE,XPL) to plot for each time sample. For further details on the Stanford Plot, the reading of [2] is highly recommended.

When focusing on integrity, though, the classical Stanford Plot is not always conservative. Indeed, using all in view satellites to measure integrity over-bounding capability may lead, for instance, to a situation in which a specific integrity loss in one or more satellite may be mitigated by other “well-bounded” line of sight, so that the net effect at position domain would not be appreciated. Furthermore, there is no obligation for the users to use always all available satellites, since for instance in some cases, some satellites in view may have been discarded because of a wrong tracking. Those users may have big discrepancies in performances with respect to others.

To overcome the above limitations, a new concept for SBAS integrity validation is presented in this work, consisting of a modification of the Stanford Plot. Detailed ESA studies on the transfer of integrity between pseudo-range and position domains [3], have led to the introduction of this specific kind of representation technique, which is able to provide a strong evidence of the robustness of an SBAS system with respect to integrity bound provision for all possible satellite geometrical conditions. This new representation is then exclusively focused on Integrity (versus the Integrity, Availability, and Accuracy information of the bi-dimensional nominal Stanford Plot).

During the SBAS Interoperability Working Group meeting celebrated in Madrid (Spain) in March 2005, it was suggested to call this new Integrity representation as the “Stanford-ESA Integrity Diagram”. The Stanford-ESA modified Integrity Diagram concept is described in this paper, and a quick and simple algorithm to compute this diagram is provided in the appendix, together with a source code example of its implementation in C. Results for the European Geostationary Navigation Overlay Service (EGNOS) system are also shown in this paper.

The Stanford-ESA integrity diagram has been included in the GNSS Supervisory Authority (GSA) recommendations for the EGNOS safety certification as a main tool for the integrity assessment in the position domain.

THE STANFORD-ESA INTEGRITY DIAGRAM

The Stanford-ESA Integrity Diagram, as the name itself indicates, is a modification of the well known Integrity-Availability-Accuracy 2D histogram proposed by the WAAS laboratory of the Stanford University, commonly known as “Stanford Plot”. This new Integrity Diagram concept proposes exactly the same representation technique, but introducing a modification in the data to be used as input source to build the graph, which focuses exclusively on integrity.

The “all geometries” Stanford-ESA Integrity Diagram

Instead of representing each (XPE, XPL) pair for an all-in-view situation, it does that, at every second, for all the combinations of satellites from 4 to all-in-view. Moreover, the right of diagonal is modified as follows:

- If a MI happens, all the geometries on the right of diagonal are plotted with the same colour, and with a common symbol (square, star, triangle,...). And the colour indicates the epoch relative to the time interval recorded in the file, in the colour bar of the plot.
- The previous rule applies only to the last 10 epochs with MIs. All the other epochs with MIs (if it happens) will be display as in the left side. Nevertheless, this situation should be extremely rare and will indicate an important anomaly from the receiver or system.

The aim of previous rules is to identify the sets of geometries associated to the same MI and, especially, to provide some information about the time correlation of such MIs. In this way, the all-geometries diagram provides a quick and clear view of the system performance for the all possible geometries (not only for all-in-view). And, at the same time, the display is enhanced with time information for the unsafe geometries (i.e., with MIs), which are clustered by epochs using different symbols. Finally, at the bottom of the plot it is written the “number of epochs and geometries with XPE>XPL”, which summarizes the simple pass/fail criterion: no points on the right of diagonal (for any geometry at any epoch).

Four examples of all geometries diagrams are given in figure 2. The first row shows two examples of diagram without MIs. Notice that, although the figure at left has no MIs, the pattern is quite disperse, with several points close to the diagonal. The figure at right shows the expected performance for a safety system.

The diagrams of second row do not fulfill the integrity criteria, having geometries with MIs. In the left side diagram only two epochs are involved, labelled by blue-stars and red-squares. Such epochs are far in time. The first epoch (dark blue-stars) corresponds to the beginning of the file. The second one (red-squares) happened at the end of data collecting period. Thence, such MIs are independent in time. The figure at bottom right shows a diagram with several geometries having MIs which involve four epochs. The MIs happen basically in three different periods of times: At the beginning of the file (dark blue-circles), at the middle (light green stars and triangles) and at the end of file (red squares). Notice that the epochs labeled with stars and triangles are very close in time (both have the same color).

NOTE: The underbounding situations indicated in this article are included here for the sake of illustrating the principle of the Stanford-ESA concepts proposed and do not correspond to the EGNOS system real performances.

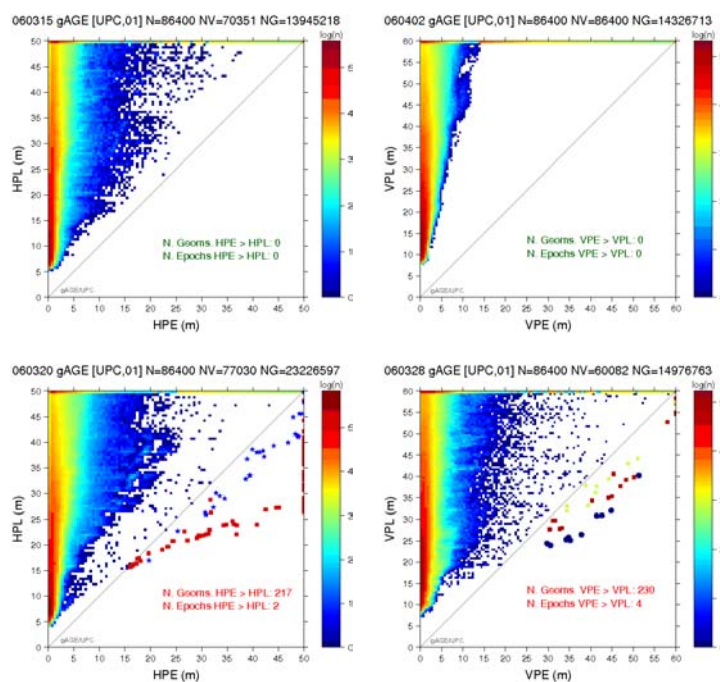


Figure 2: Four artificially generated examples of all geometries Stanford ESA Integrity Diagram. The number of epochs (N) in the measurement file, the number of epochs with valid navigation solution (NV) and the number of computed geometries (NG) are shown at the top of each diagram.

The idea of considering all possible subsets of satellites is not new, it is widely used in Receiver Autonomous Integrity Monitoring (RAIM) for satellite fault detection and exclusion [4]: The receiver explores all the

combinations of n satellites in view ($n > 4$, for detection and $n > 5$ for exclusion) where n is the number of satellites range measurements used in the position solution. For each subset of satellites, it performs a least square solution of position and clock offset based on pseudorange measurements to the chosen satellites. A statistic test for fault detection is then derived as a function of the range residuals.

Other example is the Position Domain Monitoring (PDM) concept introduced in Local Area Augmentation System (LAAS) [5], where a remote receiver hosting the PDM derives position solutions using all visible satellites (approved by the LAAS Ground Facility) and all reasonable subsets of these satellites. These position solutions are compared to the known (surveyed) PDM antenna, and errors exceeding the detection threshold are alerted.

In our case, the target is not to exclude the faulty satellites or to warn the users in a real time monitoring. The target is to have a tool to perform a strong SBAS Position Domain Integrity Validation, in such a way that, if no over-bounding is detected for any geometry at the user level, thence it can be assured that the integrity margins are safe.

This cannot be fully assessed with the classical Stanford plot, as it is shown in next figures 3 and 4:

Figure 3 provides a clear example about the limitations of the classical Stanford plot, against the all geometries Stanford-ESA Integrity Diagram. The first row shows the results for the non-safety of life ESTB system and the second row the results for the real EGNOS signal (see also figure 5). The figures at the left show the classical Stanford Plots. The figures at the right show the associated all geometries diagrams. While the Stanford Plot at the top looks like the best performance, with availability of 100%, is the Stanford-ESA Integrity Diagram at the bottom who has the best performance.

Figure 4 shows another example of how optimistic can the classical Stanford Plot be. The figure at the left show the classical Stanford Plot without having MIs. The figure at the right shows a very unsafe situation with a huge number of geometries having MIs. This result corresponds to an specific critical day of the non-safety of life ESTB system in Canaries Islands (North-West of Africa).

As stated later on this article, these examples clearly illustrate author's conviction on the importance of complementing daily user/pseudorange bounding routine performance tests with Stanford-ESA tests around Europe sites, so that all geometries are systematically assessed, confirming hopefully system margins are sound in all circumstances. This maybe of particular interest upon major identified GPS or ionospheric threat events, to confirm user domain margins remain constant upon a system maintenance correction, etc.

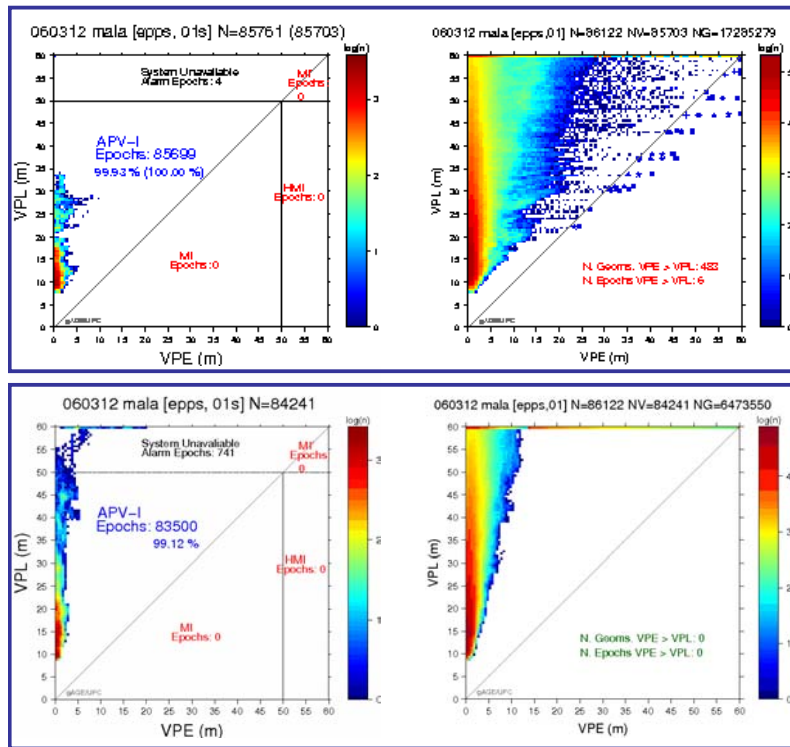


Figure 3: Two examples comparing the performance of the Stanford Plot and the all geometries Stanford-ESA Integrity Diagram. Each row shows the Stanford Plot and its associated Stanford-ESA Integrity Diagram. The first row corresponds to the non-safety of life EGNOS Test Bed (ESTB) system (PRN 120) and the second row to the real EGNOS signal (PRN 126).

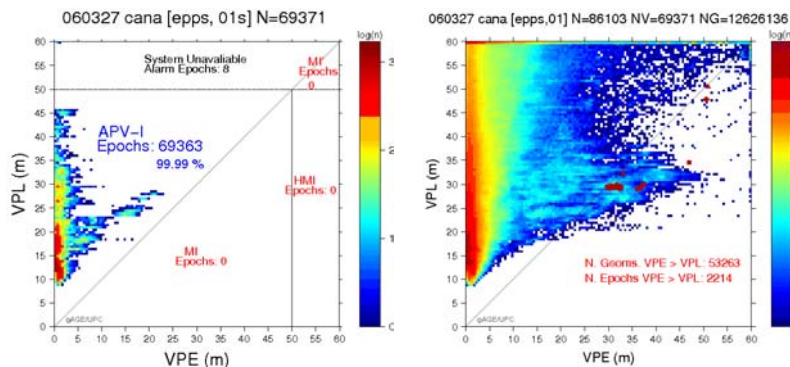


Figure 4: Example of how conservative can the classical Stanford Plot be. No MIs appear in the Stanford Plot, while a very unsafe performance is displayed in the all geometries Stanford-ESA Integrity Diagram. Results computed from the non-safety of life ESTB system signal (PRN 120).

COMPUTATION ALGORITHM

A simple sequential/recursive algorithm to compute the XPE and XPL for all possible geometries of satellites in view with very low storage and CPU requirements is provided in the appendix. An example of its implementation in C software is also shown.

The algorithm starts up from the normalized geometry matrix and measurement vector for all satellites in view with valid differential corrections available. It excludes one satellite and computes the XPE and XPL; then iterates recursively from previous matrix and vector up to when only 4 satellites remain (i.e., recursively going over one branch in the “combinations tree”). Next, it comes back to the starting matrix, excludes another satellite and recursively iterates again. This scheme is sequentially applied over all the possible branches, covering all combinations of satellites.

The strategy has low storage requirements. Only a single geometry matrix \mathbf{G} and measurement vector \mathbf{y} have to be stored (at most) at the same time (see comment 2 in the appendix). No combination is computed twice.

The previous algorithm has been implemented in BRUS [6], the software developed by gAGE/UPC to process and analyze the SBAS data. It has also been incorporated in a testing version of the Global Monitoring System (GMS) [7] that is computing daily the EGNOS performance.

The computation of the “all geometries” Stanford-ESA diagram for the 24h data set at 1Hz of figure 3 (at top), involving 17.285.279 geometries required only 1 min. and 2 sec. of CPU additional to the computation of “one single” geometry for each epoch (see table 1).

<i>Computation</i>	<i>Time</i>
One geometry for each epoch (to make the standard Stanford plot)	1 min 50 sec (85.703 geometries)
All geometries for each epoch (to make the All Geometries Stanford-ESA diagram)	2 min 52 sec (17.285.279 geometries)
Total additional time	1 min 2 sec

Table 1. Computation time over a standard PC with LINUX (Pentium 4, CPU 3.0 GHz).

An assessment of the additional computation time required to build the 24h Stanford-ESA Diagram relative to the Stanford plot in function of the number of satellites (continuously) in view is given in table 2. About the 50% of CPU time is devoted to the internal loops, vector assignation ... and the other 50% to compute the navigation solution. These results have been computed running the software code of the appendix over a standard PC with Linux (Pentium Dual Core CPU 3.4 GHZ). Two different approaches have been used to compute the processing time:

- 1) Using the source code given in the appendix in a single processor environment.
- 2) Optimizing the software code for multicore processors (in the table is shown the behavior in a dual core processor).

The results show that although the number of geometries increases with 2^N , when the number of satellites N is large, the computation of a full 24h data Diagram is feasible having up to 20 satellites in view (with the current computation capabilities of a standard PC). Notice that doubling the number of cores, the CPU time is reduced by half (equivalently, to process one more satellite with the same time).

Previous results suggest that this technique will be feasible for the combined GPS and Galileo constellations, taking into account the computation capabilities evolution (faster clocks and multicore processors).

Number of Sat. in view	Number of Geometries	Processing time	Processing time using dual core CPU features
8	14.083.200	26 sec	24 sec
10	73.267.200	1.5 min	1.5 min
12	328.060.800	7 min	3.6 min
15	2.781.388.800	1 h	29 min
17	11.252.563.200	4 h	2 h
18	22.563.878.400	9 h	4 h
20	90.480.240.000	37 h	18 h
21	1.8 e+11	3 days	37 h
22	3.6 e+11	6 days	3 days
25	2.8 e+12	45 days	23 days
N	$\sim 86400 \times 2^N$	$\sim 2^{(N-19.5)}$ days	$\sim 2^{(N-20.5)}$ days

Table 2. Additional time required to compute the Stanford-ESA Integrity Diagram relative to the Stanford Plot. Tests done over a standard PC with LINUX (Pentium Dual Core CPU 3.4 GHz).

First column stands for the number of satellites in view. Second column the number of geometries involved in 24h. Third column the processing time without using the dual core features. Fourth column, a multithread software code optimized for multicore processors.

EGNOS MEASUREMENT RESULTS

Stanford-ESA performance results with real EGNOS signal in space were computed for a set of stations covering a wide range of locations in Europe. These results, provided here for illustration, were obtained on March 12th 2006 with PRN 126 EGNOS transmission.

Figure 5, show the all geometries Stanford-ESA Integrity Diagrams in the Vertical domain for 6 fixed sites which coordinates are given in table 3. All data sets were collected at 1Hz.

Results clearly reveal excellent EGNOS integrity margins for all geometries. The integrity margins remain large, and for all samples, it is confirmed that the computed protection levels do always bound the user position error.

Station	Location	Latitude	Longitude
dlf5	Delft (The Netherlands)	51.99	4.387
fuc2	Fucino (Italy)	41.98	13.60
mala	Malaga (Spain)	36.68	-4.49
scan	Scanzano (Italy)	37.91	13.36
tlse	Toulouse (France)	43.54	1.39
upc3	Barcelona (Spain)	41.27	1.98

Table 3. Site coordinates.

A regular daily monitoring of the Stanford-ESA diagrams at several European sites is a very good complement to daily accuracy, availability and continuity performance measurements and could better help to understand the actual EGNOS safety margins, allowing, through real data, to confirm the designed safety margins. This monitoring can be done by a plot such as of figure 6, derived from the Stanford-ESA Integrity Diagram. Each point in this plot represents the “worse ratio of XPE/XPL” for a given site in 24 hours. Thence, when this value is under 1, it means that at the user domain worse-ever possible case, there is no situation in which the error overcomes the protection level, on this day and for such station.

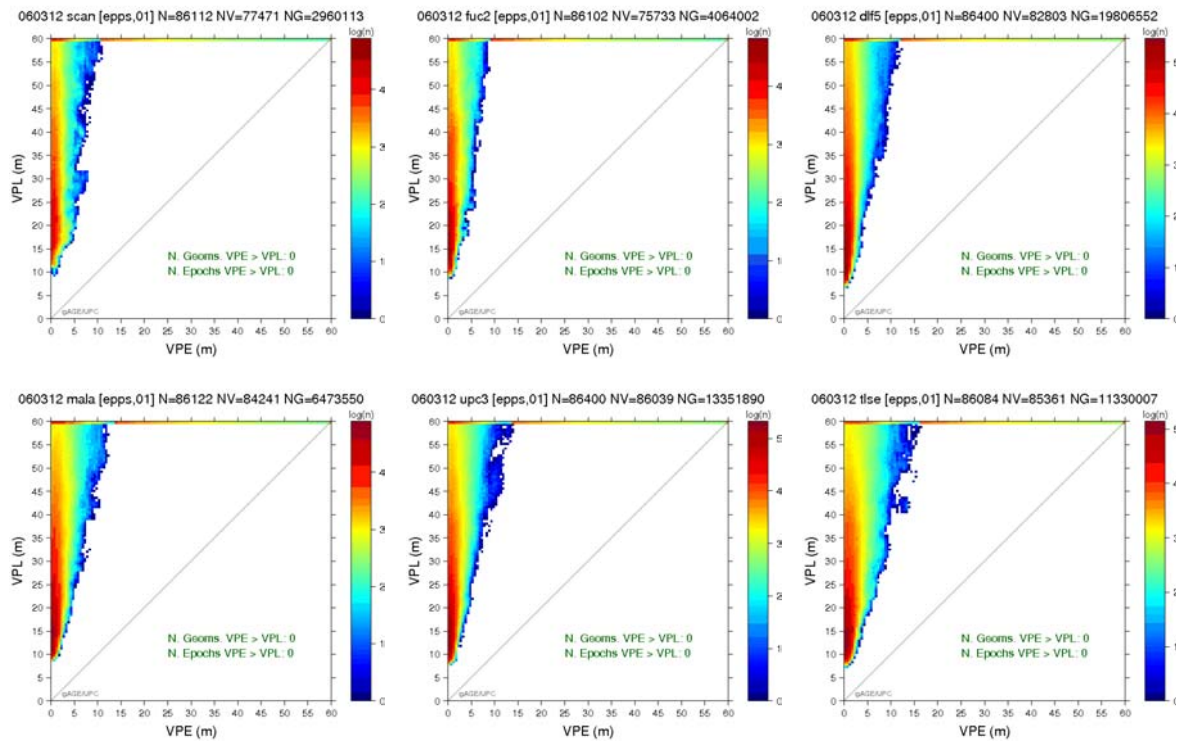


Figure 5: Vertical component: All-Stanford-ESA diagram for 6 sites covering a wide range of locations in Europe at March 12th 2006 (EGNOS, GEO PRN 126). The coordinates are given in table 3.

The plot at top of figure 6 shows how the above mentioned very exigent criterion is fulfilled, in the vertical domain, by all the monitored sites of a wide network of stations in Europe for more than three consecutive months of real EGNOS signal (GEO PRN 126). The same plot, but with the non-safety of life ESTB system signal (GEO PRN 120), is shown at bottom for comparison, illustrating the sensitivity of this plot to the system integrity.

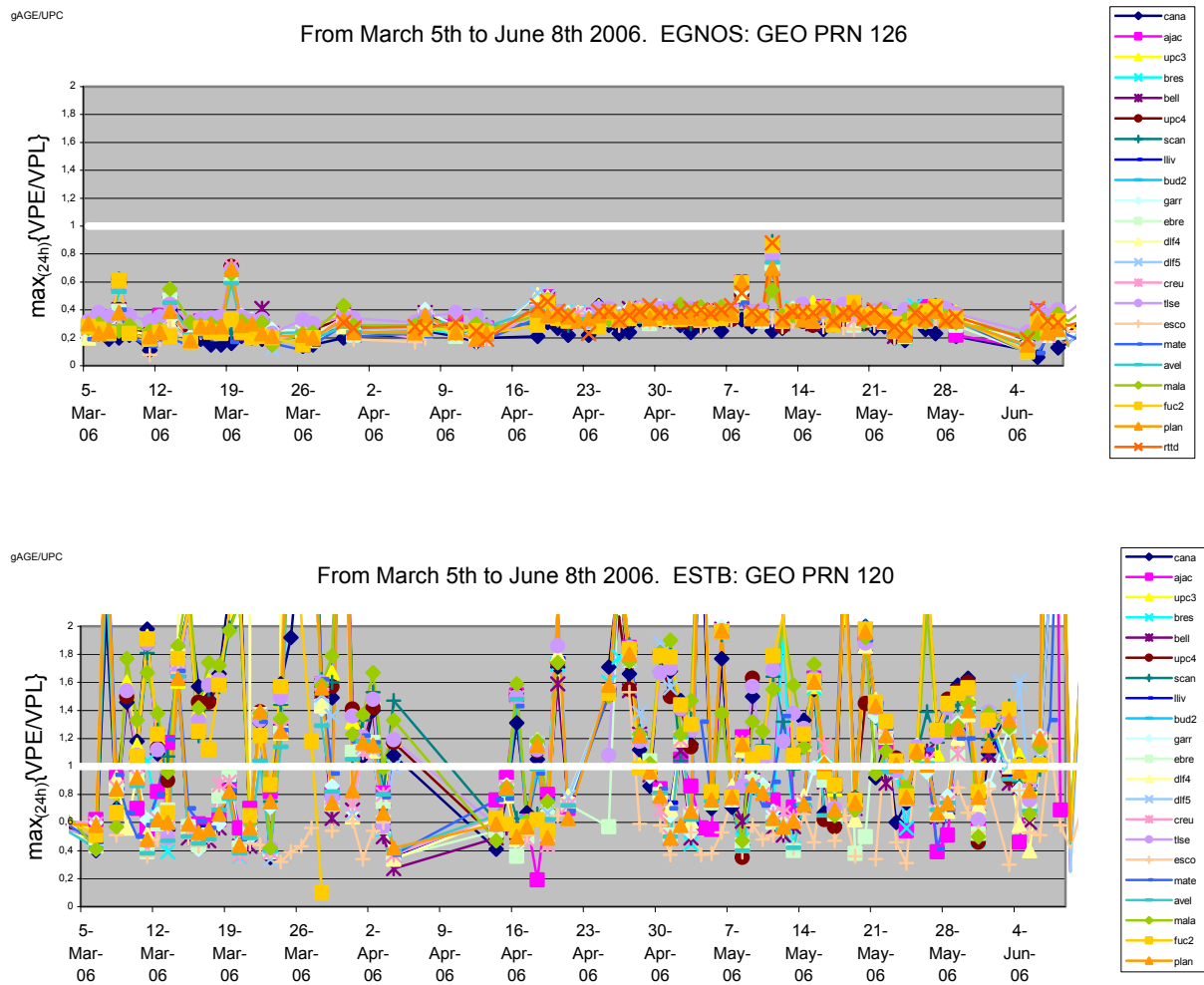


Figure 6: The Worse daily VPE/VPL ratio computed for a network of stations covering a wide range of locations in Europe, from March to June 2006. The plot at top corresponds to the true EGNOS signal (GEO PRN 126). The plot at bottom corresponds to the non-safety of life ESTB system signal (GEO PRN 120).

CONCLUSIONS

A new concept for SBAS integrity validation is presented in this work, consisting of a modification of the well known and extensively used Stanford diagram.

Results show how “unsafe system performances” are amplified by the Stanford-ESA Integrity Diagram (by running for all geometries), proving its ability to better discriminate between safe and unsafe systems, than the classical Stanford Plot. Obviously, with the Stanford-ESA Integrity Diagram the results for accuracy and availability do not have any meaning and the analysis shall only focus on integrity. Indeed, showing that in this user level domain, there is no situation for any possible geometry in which the error overcomes the protection level, then this would be the best experimental guarantee that at user domain, for a specific location and epoch, no over-bounding is incurred. In this way, the computations with real measurement shows that the EGNOS integrity margins remain safe when the Stanford-ESA integrity diagram is computed, a certainly very positive result.

The Stanford-ESA Integrity diagram is considered by the authors to be a powerful tool for safety analysis, since it may easily be applied to real data, and without significant CPU or storage requirements. It could be suggested, for instance, that involved European civil aviation and related certification authorities, could apply, during EGNOS initial operations, daily collection campaigns at well selected European sites, computing, among other parameters, the Stanford-ESA Integrity Diagrams proposed here, and reporting on the actually obtained margins. This could certainly be of support to the EGNOS Safety case and could better help to understand the actual EGNOS safety margins. It needs no explanation that the same technique can also be applied to other SBAS systems and in the future to Galileo.

With the mathematical approach proposed by the authors in the Appendix, the Stanford-ESA diagrams may be computed in real time at any given location. It could then be suggested as a complementary real time monitoring for the SBAS operators, allowing the identification in time of possible problems at user level. Finally, we believe these concepts may also have an interest in the area of Ground Based Augmentation Systems (GBAS).

ACKNOWLEDGMENTS

We would like to thank the European Space Agency, Eurocontrol, The Institut Cartografic de Catalunya (ICC) and the IGS community for providing us with the ground receiver data. The contributions from gAGE/UPC to this work are done in the frame of EGNOS SIS validation activities under Eurocontrol contract, with the additional support of the Spanish projects IBER-WARTK and WARTKATNET.

REFERENCES

- [1] SARPS Amendment 77, Annex 10 to the Convention on International Civil Aviation, Aeronautical Telecommunications: International Standards and Recommended Practices, Volume 1, Radio Navigation Aids, November 2002.
- [2] WADGPS Laboratory (Stanford University). "WAAS Precision Approach Metrics. Accuracy, Integrity, Continuity and Availability," <http://waas.stanford.edu/metrics.htm>. October 1997.
- [3] A. Tadjine. "WP 103.2: Integrity at User Level (Part II)". Issue 1, Rev. B. Ref. GMV-EGNOSAE-TN-007/04. Dated 17/01/2005.
- [4] B.W. Parkinson. Global Positioning System: Theory and Applications, Vol II. Progress in Astronautics and Aeronautics. Vol 164. Published by the Institute of Aeronautics and Astronautics, Inc. 1996.
- [5] LAAS Position Domain Monitor Analysis and Failure-Test Verification. J. Lee, S. Pullen, G. Xie and P. Enge, Stanford University, Stanford, CA. 21st International Communications Satellite Systems Conference and Exhibit, Yokohama, Japan, Apr. 15-19, 2003
- [6] Basic Research Utilities for SBAS. M. Hernández-Pajares, J. M. Juan and J. Sanz. gAGE/UPC 2003. <http://gage9.upc.es>.
- [7] GMS Architectural Design and Detailed Design Document, M. Hernández-Pajares, J. M. Juan and Jaume Sanz. gAGE/EEC, 2004. EUROCONTROL.

APPENDIX: COMPUTATION ALGORITHM

The basic linearized GPS measurement equation $y = \mathbf{G}x$ with weighting matrix \mathbf{W} can be transformed into the "normalized" system $y_w = \mathbf{G}_w x$ with weighting matrix being identity matrix, by introducing the normalized vector $y_w = \sqrt{\mathbf{W}} y$ and the normalized matrix $\mathbf{G}_w = \sqrt{\mathbf{W}} \mathbf{G}$. (Notice that \mathbf{W} is a diagonal matrix).

The algorithm:

1) Let's be $y_w^* = G_w^* x$ the system after removing the equation of one satellite, and

$$function_nav_sol(y_w, G_w, k, y_w^*, G_w^*, HPE^*, VPE^*, HPL^*, VPL^*)$$

the function that:

- Removes the row k of matrix G_w , and the component k of vector y_w , (of orders $N \times 4$ and N , respectively) providing the matrix G_w^* and vector y_w^* (of orders $(N-1) \times 4$ and $N-1$, respectively).
- Computes the XPE^* and XPL^* from the system $y_w^* = G_w^* x$.

That is:

INPUT: k, y_w, G_w

OUTPUT: $y_w^*, G_w^*, HPE^*, VPE^*, HPL^*, VPL^*$

2) Let's be $nsat$ the number of satellites in view with valid differential corrections available:

For each epoch (having a sample of $nsat$):

 Compute the XPE and XPL from the system $y_w = G_w x$ (solution with all satellites)

 for $k1=1, nsat$

$$function_nav_sol(y_w, G_w, k1, y1_w, G1_w, HPE1, VPE1, HPL1, VPL1)$$

 for $k2=k1, nsat-1$

$$function_nav_sol(y1_w, G1_w, k2, y2_w, G2_w, HPE2, VPE2, HPL2, VPL2)$$

 for $k3=k2, nsat-2$

$$function_nav_sol(y2_w, G2_w, k3, y3_w, G3_w, HPE3, VPE3, HPL3, VPL3)$$

.....

(up to only 4 satellites remain)

 endfor

 endfor

endfor

Note: At each iteration, it must be only saved the geometry matrix and measurement vector y_w^*, G_w^* as an input for the next one.

Additional comments:

Comment 1:

function_nav_sol($y_w, G_w, k, y_w^*, G_w^*, HPE^*, VPE^*, HPL^*, VPL^*$)

$$x^* = (G_w^{*t} G_w^*)^{-1} G_w^{*t} y_w^* \quad ; \quad P_x^* = (G_w^{*t} G_w^*)^{-1}$$

where $G_w^{*t} G_w^*$ is a 4x4 definite positive matrix.

If the observation matrix G is given in “East, North, and Vertical” coordinates (see appendix E and J [RD.1]),

thence:

$$\begin{aligned} HPE^* &= \sqrt{x_1^{*2} + x_2^{*2}} \\ VPE^* &= x_3^{*2} \\ HPL^* &= k_H \sqrt{\frac{P_{11}^{*2} + P_{22}^{*2}}{2} + \sqrt{\left(\frac{P_{11}^{*2} - P_{22}^{*2}}{2}\right)^2 + P_{12}^{*2}}} \\ VPL^* &= k_V P_{33}^* \end{aligned}$$

Comment 2:

A more compact version of the algorithm involving only the storage of single matrix G_w and vector y_w could be achieved by using a recursive function:

function_nav_recursive ($k, kmask, depth, nsat, yw, Gw$)

Being:

- k , a vector that stores the equivalent $k_1, k_2, k_3...$ of the first version of the algorithm.
- $kmask$, a vector which implies a mask for the satellites (so for the rows of G_w and the components of y_w), having a value 1 when a satellite is selected for the navigation solution and 0 when it is deselected.
- $depth$, the profundity of the present branch inside the tree.

- nsat, the total number of satellites in view and with valid differential corrections in the present epoch.
- yw and Gw, the normalized vector and geometry matrix.

The C “pseudocode” of the function would be:

```
function_nav_recursive (k, kmask, depth, nsat, yw, Gw) {  
    // To call the function in main:  
    // k[0]=0;  
    // for (i=0;i<nsat;i++) {  
    //     kmask[i]=1;  
    // }  
    // function_nav_recursive (k, kmask, 1, nsat, yw, Gw);  
    compute_nav_solution (kmask, nsat, yw, Gw, HPE,VPE,HPL,VPL);  
    if ((nsat-depth)>4) {  
        for (i=k[depth] ; i<=(nsat-depth-1) ; i++) {  
            k[depth+1]=i;  
            kmask[i+depth]=0;  
            function_nav_recursive (k, kmask, depth+1, nsat, yw, Gw);  
            kmask[i+depth]=1;  
        }  
    }  
}
```