# Error bound optimization using second order cone programming

Juan Blanch, Todd Walter, Per Enge. *Stanford University*

**ABSTRACT**

In safety of life applications using satellite navigation, the Protection Level (PL) equation translates what is known about the pseudorange errors into a hard bound on the positioning error (the Protection Level). The current PL equations for Satellite Based Augmentation Systems are based on Gaussian statistics: all errors are characterized by a zero mean Gaussian distribution which is an upper bound of the true distribution. This approach is very practical: the calculations are simple and the receiver computing load is small. However, when the true distributions are far from Gaussian, such characterization forces an inflation of the protection levels that damages performance. Also, in the certification process, it is very difficult to agree on a given distribution when the statistics are gathered from a multitude of situations (like differing elevation angle in the case of multipath), or when the process involved has large deviations from a Gaussian model (stormy ionospheric behavior). With the development of new optimization methods and the increasing computing power, it is worthwhile exploring new ways of calculating integrity error bounds.

In this paper we present a way of computing optimal protection levels when the errors are characterized by a Gaussian random component and a bias that is linearly constrained. We show that the minimization of the protection level can be cast as a second order cone program (SOCP), and that this particular structure allows a quick computation. As an example, this algorithm is applied to a dual frequency Wide Area Augmentation System where biases are sent to the user (as well as the usual Gaussian terms).

**INTRODUCTION**

Now that the next generation WAAS is being developed, it is the moment to determine which new information should be included in the new L5 messages in order to increase performance. It has been suggested [1] that sending maximum bias information through the new messages could decrease the Protection Levels, and thus increase performance. Here, we want to determine, once we have bias information, what is the best way of combining the pseudorange measurements in order to minimize the Protection Level.

In this work, *bias* designates an error that does not change with time (or very slowly) and that is susceptible to cause systematic positioning errors over time (as opposed to random errors) for a given user, or at a given location. Some examples of biases in the case of SBAS are:

- antenna bias at the reference stations caused by the differential group delay. This bias affects both the clock and ephemeris estimate and the ionospheric estimate and can reach several tens of centimeters.
- the systematic error in the geostationary pseudorange tracking caused by signal deformation [2]. Although this bias is random from user to user, it is fixed for a given user. For more details about this problem please refer to [2]. It has been suggested that even GPS pseudoranges might be biased: what is now generally accepted to be caused by multipath could actually be due to signal deformation.

In addition to the previous examples, where we only add biases to the current random errors, a more radical approach can be taken, which is to characterize all errors as biases. These biases could be subjected to linear constraints. For example, during disturbed ionospheric conditions, it is problematic to characterize the errors as being Gaussian. In this case, it leads to extremely conservative estimates of the position error and it is more natural to describe the error as a set of biases constrained by what has actually been observed. Such an approach has been suggested in [3].

Current PL equations for SBAS are based on Gaussian statistics: all errors are characterized by a zero mean Gaussian distribution which is an upper bound of the true distribution in a certain sense [4]. For a detailed explanation on the current SBAS VPL equation please refer to [5]. Here only a summary is provided. Using the SBAS messages, the user computes the inverse of a weighting matrix (where $n$ is the number of pseudorange measurements):

$$W^{-1} = C = \begin{bmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_n^2 \end{bmatrix}$$

This matrix can be interpreted as the covariance of a Gaussian overbound of the pseudorange errors. The matrix $H$ that projects the measurements onto the user position estimate is then ($G$ is the 4 by $n$ matrix describing the line of sights):

$$H = \left(G^T W G\right)^{-1} G^T W$$

This projection matrix minimizes the variance of the positioning error in each coordinate. Let us assume that the third column of G is the vertical component of the line of sight. The variance of the vertical error is:

$$\left(G^T W G\right)^{-1}_{3,3}$$

The VPL is defined as:

$$VPL = K\sqrt{\left(G^T W G\right)^{-1}_{3,3}}$$

where $K$=5.33 (corresponding to a probability of $10^{-7}$ of being outside the PL). An alternate form of the variance that will make it easier for us to modify in the next section is:

$$\left(G^T W G\right)^{-1}_{3,3} = k^T C k$$

where $k$ is the third row of coefficients of the matrix $H$.

The outline of the paper is as follows: first we will modify the VPL equation so that it includes biases, then we will show how to optimize the VPL under the new formulation using second order cone programming, and finally, as an example, we will show how we can study the full benefit of sending biases through the WAAS channel with this type of optimization.

## A VPL EQUATION INCLUDING BIASES

First we give a new error characterization. Every pseudorange error $\varepsilon_i$ is characterized as a normal distribution $N(\mu_i, \sigma_i)$, where the only information known about $\mu_i$ is $|\mu_i|<\beta_i$. Under this characterization, and assuming that the coefficients applied to the measurements to determine the vertical position are labeled $k_i$, then a conservative vertical error bound is given by:

$$VPL = K\sqrt{k^T C k} + \sum_{i=1}^{n} |k_i \beta_i|$$

There are ways to take advantage of the fact that a biased solution has asymmetric tails, but they will not be considered in this work.

In the previous equation, the coefficients $k$ have to be such that when the measurements have no error, we retrieve the true position solution. Let us now consider the matrix $H$ that projects the pseudorange measurements onto the position solution $x_{est}$ ($k$ is one row of $H$). We break down the pseudorange into the true pseudorange $y$ and the error. We have:

$$\begin{aligned} x_{est} &= H\left(y + error\right) \\ y &= Gx \end{aligned}$$

Without error and for any position we want:

$$x_{est} = HGx = x$$

As a consequence, $HG=I$ and:

$$G^T k = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}$$

To be valid, any set of coefficients $k$ needs to satisfy this linear constraint.

## VPL OPTIMIZATION

In this section we show how to obtain, for a given geometry, the minimal error bound under the error characterization presented in the previous section.

The problem can be stated as:

$$\begin{aligned} &\text{minimize} && K\sqrt{k^T C k} + \sum_{i=1}^{n} |k_i \beta_i| \\ &\text{subject to} && G^T k = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \end{aligned}$$

As stated, this problem does not have a recognizable structure. The first step consists in expressing differently the quantity to be minimized. Let us define $A$ and $b$ as:

$$A = \begin{bmatrix} I_n \\ -I_n \end{bmatrix},\ b = \begin{bmatrix} \beta \\ \beta \end{bmatrix}$$

We have:

$$\sum_{i=1}^{n} |k_i \beta_i| = \max_{A\varepsilon \le b}\left(k^T \varepsilon\right)$$

The notation $A\varepsilon \le b$ has to be interpreted component wise. The problem is now:

$$\begin{array}{ll} \text{minimize} & \max\limits_{A\varepsilon\le b} \left(K\sqrt{k^T C k} + k^T \varepsilon\right) \\ \text{subject to} & G^T k = \begin{bmatrix}0 & 0 & 1 & 0\end{bmatrix} \end{array}$$

Under this form, the problem looks more complex than in the original form. This transformation has two purposes: first, to show that the problem can be generalized to more complex constraints on the biases; second, we show in the Appendix that this problem is equivalent to:

$$\begin{array}{ll} \text{minimize} & K\sqrt{\lambda^T A C A^T \lambda} + b^T \lambda \\ \text{subject to} & G^T A^T \lambda = \begin{bmatrix}0 & 0 & 1 & 0\end{bmatrix} \\ & \lambda \ge 0 \end{array}$$

This transformation can be interpreted as the change of variable: $A^T \lambda = k$.

Before treating the general case, we now go over two limiting cases: *b=0*, and *C=0*. When *b=0*, it is easy to see that we end up with the current least squares problem, which has an analytical solution. With *C=0*, the minimization problem becomes a linear program (LP). Although there are no analytical solutions for linear programs, they can be very efficiently solved and LP solvers are now widespread and easily available.

It turns out that the general problem has also a standard structure known as Second Order Cone Programming [6]. To make this structure more apparent, we introduce a new variable ν and write the problem as:

$$\begin{array}{ll} \text{minimize} & \nu + b^T \lambda \\ \text{subject to} & G^T A^T \lambda = \begin{bmatrix}0 & 0 & 1 & 0\end{bmatrix} \\ & \lambda \ge 0 \\ & K\sqrt{\lambda^T A C A^T \lambda} \le \nu \end{array}$$

Under this form, we can see that we are optimizing over the intersection of a simplex and a second order cone. Second Order Cone programs (SOCP) are a class of convex problems that have been extensively studied in convex optimization theory and for which there exist very efficient solvers (although not as widespread as LP solvers) that use interior point methods. These iterative solvers have the very important following properties:

- they converge in very few steps (typically less than 20)
- they provide an upper bound on the distance to optimality
- they converge in polynomial time to a given accuracy

In this work, we have not applied these solvers in an actual receiver. Instead, we have relied on a MATLAB tool to simulate WAAS user geometries. This way, we could use a MATLAB based toolbox to solve the Second Order Cone Programs. We chose to use the free MATLAB package SeDuMi [7], interfaced with YALMIP [8]. Once installed, these tools are almost transparent for a MATLAB user: there is only three new commands to learn.

The tools mentioned have been typically developed for very large numbers of variables (typically hundreds). Here, we only have tens of variables (here the number of variables is the size of the vector λ). The solver reached the optimal solution with 10 digits precision in about 10 iterations. The computational load per iteration is equivalent to a least squares problem with the number of variables of the original problem. We ran this solver on a Dell 4600 running with a Pentium 4 at 2.8 GHz and it took about .3 seconds to get an error bound for a given geometry. This is still a very large computational load for a receiver but is an acceptable load for analysis purposes.

## APPLICATION TO A DUAL FREQUENCY WAAS

In this section, we are going to evaluate the benefit of sending bias information for the next generation dual frequency WAAS. Specifically, we compare two possible computations of the VPL: one that uses the optimal *k* coefficients and another one that uses a suboptimal (but reasonable) least squares algorithm to determine *k*. This study was done using the MATLAB based service volume analysis tool MAAST. This software evaluates the performance of WAAS by computing the protection level for users placed on a regular grid over a given period of time [9].

For this simulation we have assumed:

- a constellation of 24 optimal GPS satellites (which is the constellation specified by the MOPS) and 4 Geostationary satellites (POR, AOR-W, N107, N133)
- that all satellites are dual frequency L1-L5 so there is no ionospheric error other than the uncertainty on the ionospheric delay estimate;
- the current network of 25 WAAS reference stations.

For the Geostationary satellites, the sigma values corresponding to the uncertainty on the clock and the ephemeris are no longer inflated to account for the biases. Instead, a maximum bias of 5 meters is sent for each of them.

We have already outlined how the optimal algorithm was implemented. Here we describe a suboptimal algorithm. The VPL equation is still [eqref], what changes is the determination of $k$, which is no longer optimal. A reasonable way to account for the biases is to modify the matrix $C$ by adding a bias term for the geostationary satellites:

$$\sigma^2_{geo,i,\mathrm{mod}} = \sigma^2_{geo,i} + \left(\frac{bias}{K}\right)^2$$

Then the matrix $H$ is defined as:

$$H_{\mathrm{mod}} = \left(G^T C^{-1}_{\mathrm{mod}} G\right)^{-1} G^T C^{-1}_{\mathrm{mod}}$$

$k$ is then the third row of $H_{\mathrm{mod}}$.

The simulation was carried for users situated in a grid covering CONUS every 2.5 degrees during the course of a day and every 30 minutes. The VPL plots show the 95% percentile of the VPL at each location.

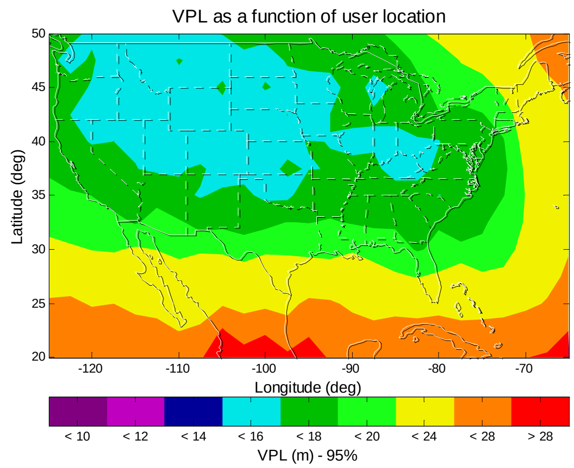


***Figure 1.*** 95% of the VPL using the optimal algorithm (SOCP)

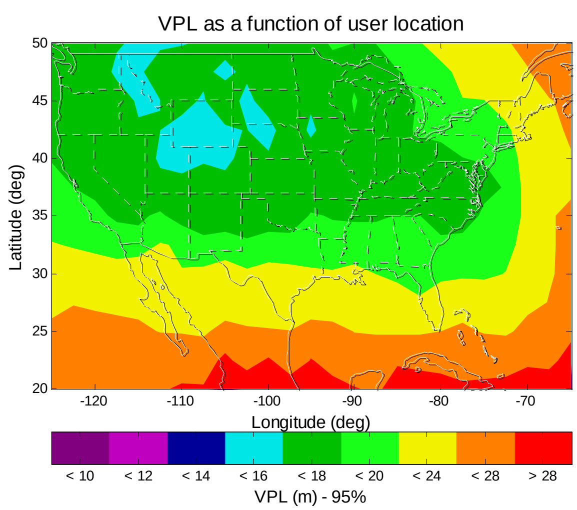


***Figure 2.*** 95% of the VPL using the suboptimal algorithm

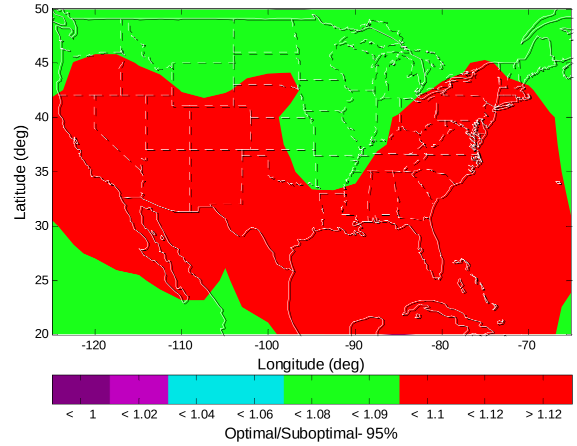


***Figure 3.*** 95% of the ratio between suboptimal and optimal algorithm

Over all geometries, the worst ratio was found to be 11%. Although these simulation results are extremely dependent on several assumptions (on the network, the constellation, and the algorithms), they suggest that a good suboptimal algorithm might provide a performance that is close enough to optimal. The most likely use of an SOCP solver would be as an analysis tool to make sure that we choose a suboptimal Protection Level equation that is not too far from the optimal one.

## CONCLUSION

We have shown that the search of the optimal Protection Level in the presence of biased Gaussian error distributions can be cast as a Second Order Cone Program (SOCP). This result is theoretically interesting, because up to now, there was only one type of pseudorange error characterization that could be used optimally to compute the position solution (zero mean Gaussian errors). We have extended this type of errors to the case where the error is characterized by a random Gaussian component (which can be multivariate) and an unknown bias linearly constrained. Although there are no analytical solutions to SOCPs, there are excellent iterative algorithms that converge very fast, in a guaranteed number of iterations and providing a bound on suboptimality.

As an example of application of SOCPs in GPS, we have evaluated the performance of a dual frequency WAAS where bias information is sent to the user. The results suggest that it is probably not worth computing the optimal VPL, as the difference with a suboptimal algorithm does not exceed 12%. As a consequence, the most likely use of such an algorithm would be as an analysis tool to determine the best suboptimal error bound calculation in case biases are included in the VPL equation.

**ACKNOWLEDGEMENTS**

This work was sponsored by the FAA GPS product team (AND-730).

**APPENDIX**

In this section we give some elements for the proof of equivalence of the two problems:

$$\begin{array}{ll} \text{minimize} & \max\limits_{A\varepsilon\le b} \left(K\sqrt{k^T C k} + k^T\varepsilon\right) \\ \text{subject to} & G^T k = \begin{bmatrix}0 & 0 & 1 & 0\end{bmatrix} \end{array} \quad \text{(Problem 1)}$$

and

$$\begin{array}{ll} \text{minimize} & K\sqrt{\lambda^T A C A^T \lambda} + b^T\lambda \\ \text{subject to} & G^T A^T \lambda = \begin{bmatrix}0 & 0 & 1 & 0\end{bmatrix} \quad \text{(Problem 2)} \\ & \lambda \ge 0 \end{array}$$

Let us call $m_j$ the minimum of Problem $j$. This proof relies mostly on the Karush-Kuhn-Tucker (KKT) conditions [10]. First, we show that there exists $\lambda^* \ge 0$ such that:

$$\max_{A\varepsilon\le b} \left(k^T\varepsilon\right) = b^T\lambda^*$$

To prove this we write the KKT conditions of optimality. Be $\varepsilon^*$ the point where the maximum is reached. There exists $\lambda^*$ such that:

$$\begin{aligned} &k = A^T\lambda^* \\ &\lambda^* \ge 0 \\ &\lambda_i^*\left(A_{i,\cdot}\varepsilon^* - b_i\right) = 0 \end{aligned}$$

From this set of equations it is easy to see that:

$$\max_{A\varepsilon\le b} \left(k^T\varepsilon\right) = k^T\varepsilon^* = b^T\lambda^*$$

This first part of the proof shows that $m_1 \ge m_2$. For the second part of the proof, we notice that if:

$$A\varepsilon \le b\,,\quad k = A^T\lambda\,,\quad \lambda \ge 0$$

then:

$$k^T\varepsilon = \lambda^T A\varepsilon \le \lambda^T b$$

which results in $m_2 \ge m_1$.

**REFERENCES**

[1] T. Walter, J. Blanch, J. Rife. "Treatment of biased errors in SBAS", *Proceedings of The 2004 International Symposium in GNSS/GPS. Sidney, Australia,* 2004.

[2] R.E. Phelts, T. Walter, P. Enge. "Range Biases on the WAAS Geostationary Satellites", *Proceedings of the Institute of Navigation National Technical Meeting*, San Diego, CA, 2004.

[3] J. Blanch, T. Walter, P. Enge, "Ionospheric Estimation Using Extended Kriging for a Low Latitude SBAS" *Proceedings of the Institute of Navigation GNSS 2004*, Long Beach, CA, 2004.

[4] B. DeCleene. "Defining Pseudorange Integrity – Overbounding," *Proceedings of the Institute of Navigation GPS-00*. Salt Lake City, UT, 2000.

[5] T. Walter, P. Enge, A. J. Hansen. "A Proposed Integrity Equation for WAAS MOPS," *Proceedings of the Institute of Navigation GPS-97*, pp. 475-84, September 1997.

[6] M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra and Applications, Special Issue on Linear Algebra in Control, Signals and Image Processing*, 1998.

[7] J.F. Sturm, `Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones', *Optimization Methods and Software* 11-12 (1999) 625-653.

[8] http://control.ee.ethz.ch/~joloef/yalmip.msql

[9] S. S. Jan, W. Chan, T. Walter, P. Enge. "Matlab Simulation Toolset for SBAS Availability Analysis," *Proceedings of the Institute of Navigation GPS-01*, Salt Lake City, 2001.

[10] S. Boyd, L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004. p 243.