

GPS EASY Suite II

easy16—Comparing GPS Precise and Broadcast Ephemerides

KAI BORRE
AALBORG UNIVERSITY

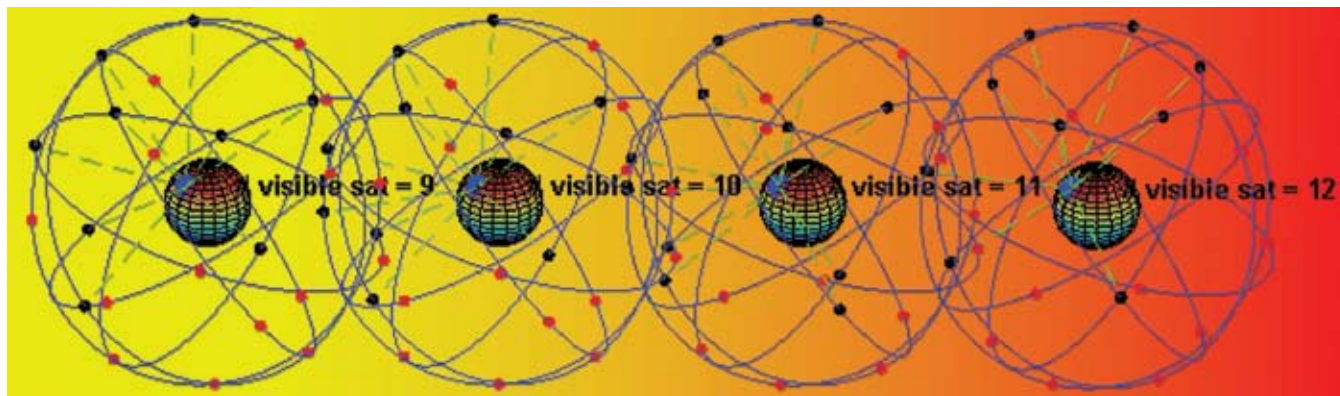


Image of GPS constellation based on public domain file from Wikimedia Commons

In this installment of the series, the author demonstrates Matlab code for estimation of ionospheric delay, multipath, and ephemeris errors.

With a given GPS receiver you get a certain positioning accuracy. So, the first question you may ask is: can I do anything to improve it?

The answer most often is: yes, go and buy a better receiver! Most ranging errors are determined by physics and you can do little to improve the situation.

However, it is interesting to analyze the information you may get from a single “one-way” range, that is, a single set of observables (P_1, Φ_1, P_2, Φ_2) related to an individual satellite seen by a single dual-frequency receiver. By so doing you also obtain an insight into the nature of the error sources — the ones that do matter and the ones that can be modeled or eliminated.

From the ephemeris of the tracked satellite we can compute the elevation angle, the ionospheric delay is estimable because the data originate from a dual-frequency receiver, the tropospheric delay is computed from a standard model, and the remaining errors we call *multipath* errors.

Finally, we estimate the orbital errors by comparing satellite positions computed from broadcast and precise ephemerides. The data sample illustrates these different error contributions very well, their variation over time, and, therefore, their dependency on elevation angle.

We select a specific one-way observation set that was acquired at a station south of Aalborg on January 19, 2007. (This is from the data set already used in easy15, published in the January/February 2010 issue of *Inside GNSS*.) We investigated PRN14 in the full session length, that is, about 42 min-

utes — including the 95 epochs that we omitted in the easy15 discussion because of missing data at either the rover or the master, or repeated estimation of ambiguities.

The subplots in **Figure 1** depict the satellite *elevation angle*, which varies from 18 to 33 degrees, and one-way errors in *ionosphere*, *troposphere*, and *multipath*.

When dealing with the ionospheric delay I , we assume that we have both pseudorange and carrier phase observations on L_1 and L_2 at our disposal. We estimate the delay I according to the following matrix equation:

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & \lambda_1 & 0 \\ 1 & \alpha & 0 & 0 \\ 1 & -\alpha & 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \rho^* \\ I \\ N_1 \\ N_2 \end{bmatrix} = \begin{bmatrix} P_1 \\ \Phi_1 \\ P_2 \\ \Phi_2 \end{bmatrix} + \mathbf{e}. \quad (1)$$

As usual we define the constant $\alpha = (f_1/f_2)^2 = I_2/I_1$. First we estimate the ambiguities on L_1 and L_2 — N_1 and N_2 , respectively — as reals; incorrect values for N_1 and N_2 just change the level for I . Next we estimate I alone as

$$I = (-\lambda_1 N_1 + \lambda_2 N_2 + \Phi_1 - \Phi_2) / (\alpha - 1). \quad (2)$$

For the tropospheric delay we use the Goad-Goodman model. The delay T ranges from 7.4 m to 4.2 meters. If the satellite passed zenith, the corresponding delay would have been about 2.5 meters.

Multipath describes the situation where signals coming from the satellite propagate along several paths to the receiver.

er antenna. The main part of the signal radiates directly from the satellite, but part of the signal is reflected from surfaces near the receiver.

Multipath occurs when the signal arrives at the antenna from these reflected surfaces in addition to the line-of-sight source. The reflected signal is phase-shifted with respect to the original transmission and appears as additive noise at the antenna.

Multipath depends on satellite geometry and the antenna environment, which makes multipath difficult to model. For long observation periods — 24 hours or more — the multipath effects are partly reduced by averaging. However, observation periods usually last for only a few hours and often much less; this is why multipath is a problem.

Because the antenna locations are different, the multipath signa-

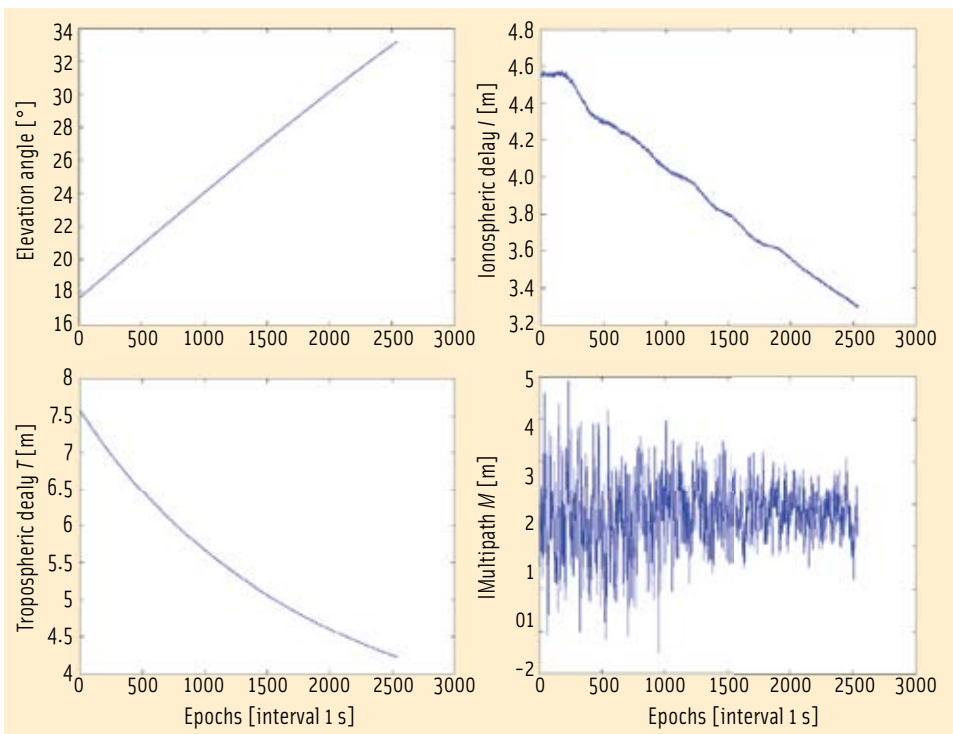


FIGURE 1 One-way errors for PRN14



ture at each antenna is unique and the error is not common mode.

Let the geometrical distance between satellite and receiver be ρ , let I be the ionospheric delay, and M the code multipath including receiver noise. The pseudoranges observed on L_1 and L_2 for PRN14 may then be expressed as

$$P_1 = \rho + I_1 + M_1 \tag{3}$$

$$P_2 = \rho + I_2 + M_2. \tag{4}$$

Remaining errors are identified as multipath and receiver noise — similarly for the phase observations:

$$\Phi_1 = \rho - I_1 + \lambda_1 N_1 + m_1 = \lambda_1 \varphi_1 \tag{5}$$

$$\Phi_2 = \rho - I_2 + \lambda_2 N_2 + m_2 = \lambda_2 \varphi_2. \tag{6}$$

The multipaths m_1 and m_2 on phase observations are so small that we subsequently put $m_i = 0$. We want to find an expression for M_1 . We start by subtracting (5) from (3):

$$P_1 - \Phi_1 = 2 I_1 + M_1 - \lambda_1 N_1$$

or

$$M_1 - \lambda_1 N_1 = P_1 - \Phi_1 - 2 I_1 \tag{7}$$

and subtracting (6) from (5) yields

$$\Phi_1 - \Phi_2 = I_2 - I_1 + \lambda_1 N_1 - \lambda_2 N_2 = (\alpha - 1) I_1 + \lambda_1 N_1 - \lambda_2 N_2 \tag{8}$$

or

$$I_1 = \frac{1}{\alpha - 1} (\Phi_1 - \Phi_2) + \frac{1}{\alpha - 1} (\lambda_2 N_2 - \lambda_1 N_1). \tag{9}$$

We insert (9) into (7) and obtain

$$M_1 - \lambda_1 N_1 = P_1 - \Phi_1 - \frac{2}{\alpha - 1} (\Phi_1 - \Phi_2) - \frac{2}{\alpha - 1} (\lambda_2 N_2 - \lambda_1 N_1) \tag{10}$$

or

$$M_1 - (\lambda_1 N_1 - \frac{2}{\alpha - 1} (\lambda_2 N_2 - \lambda_1 N_1)) = \tag{11}$$

$$P_1 - (\frac{2}{\alpha - 1} + 1) \Phi_1 + \frac{2}{\alpha - 1} \Phi_2.$$

We can reasonably assume that $E\{M_i\} = 0$. The second term on the left side of (11) is a constant; so, it is possible to reduce M_1 to M_1^* such that $E\{M_1^*\} = 0$:

$$M_1^* = P_1 - \frac{\alpha + 1}{\alpha - 1} \Phi_1 + \frac{2}{\alpha - 1} \Phi_2. \tag{12}$$

Analogously, by exchanging subscripts, we have for the multipath on L_2 :

$$M_2^* = P_2 + \frac{\alpha + 1}{\alpha - 1} \Phi_2 - \frac{2}{\alpha - 1} \Phi_1. \tag{13}$$

For all epochs with all four observations P_1, P_2, Φ_1 , and Φ_2 available, we compute multipath according to Equation (12). In the present case, the average error is below two meters and noisier at low satellite elevation angles. The noisy character of the plot reflects the actual noise in any pseudorange observation.

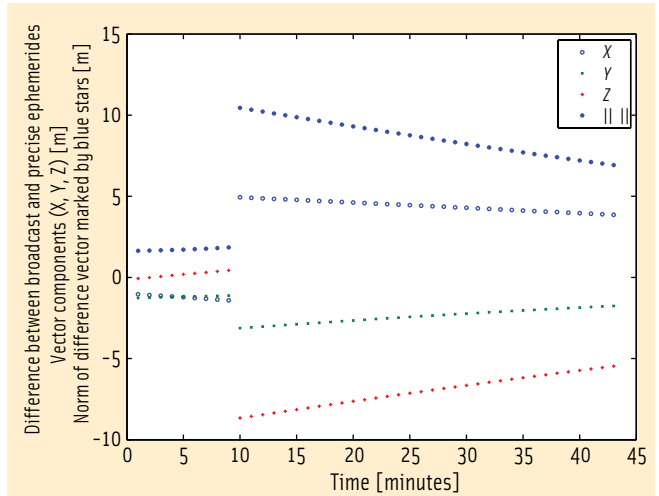


FIGURE 2 Difference in meters between precise and broadcast ephemerides for PRN14 for individual vector components (X, Y, Z). Norm of difference vector marked by blue stars

Finally we investigate the difference between precise and broadcast ephemerides. Any file containing broadcast ephemerides is specific to the site from which the satellites were tracked. Therefore, we need to identify the tracked satellites in the time period during which we want to compare the precise and broadcast ephemerides.

This can be done by inspecting the corresponding observation file in the PRN14 navigation message and getting the date and the seconds of the week; next we find the corresponding Julian Day Number (jd). We find an epoch (modulo 15 minutes) 60 minutes ahead of jd , which is counted in unit of day. (The variable jd is a real number where the integer part is number of whole days and the decimal part is made up of hours, minutes, seconds and fractions of seconds all convert into fractions of a day). Finally the GPS week number is computed by the Matlab function `gps_time`.

The precise ephemerides that we used can be found at igsb.jpl.nasa.gov/igsb/product/1410, the number 1410 being the GPS week number. The file extracts to `igr14105.sp3`, which contains precise ephemerides for January 19, 2007, in the SP3 format. The IGS SP3 files typically contain satellite positions (X,Y,Z) and clock offsets for each satellite every 15 minutes, from 0:00 hours to 23:45. (The SP3 file starts with some header lines, which for our purposes we can skip.)

We read precise orbits for the interpolation period plus three times 15 minutes ahead of the observation period and three times 15 minutes after the period, i.e., 13:15 hours, 13:30 hours, ..., 15:15 hours — in total nine sets. The precise coordinates for all tracked satellites are stored in the matrix Xp .

In the following discussion, we introduce a time scale with units of 15 minutes. We could have chosen whole minutes instead. The first observation is from 13.85 hours, and the last observation is taken at 14.55 hours. The entire observation period is 2,535 seconds = 42.25 minutes = 2.8 quarters of an hour.

A Lagrange interpolation, one for each coordinate, of at least seventh order is used to compute the actual position. The Matlab function `interp` does an $(n - 1)$ th order polynomial Lagrange

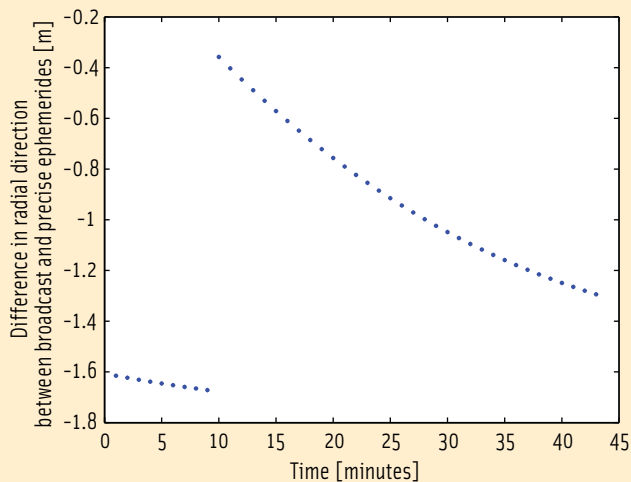


FIGURE 3 Length of difference vector between precise and broadcast ephemerides as projected onto the line of sight

interpolation. Let y be an $n \times 1$ vector of data given at the discrete times x , (x is as well an $n \times 1$ vector), (See the relevant discussion in *Matrix Analysis* by R. A. Horn and C. R. Johnson, Volume 1, pages 29–30, cited in Additional Resources at the end of this article.)

We want an interpolated precise position each minute; hence, we divide by 15 to keep units in quarters of an hour.

The first parameter in the `intp` procedure describes the abscissae for the points at which we know the satellite coordinates. The second parameter contains these coordinates, and the third parameter describes the point set at which we want interpolated values — all in units of quarter of an hour. Each coordinate is interpolated separately!

Figure 2 shows the difference in satellite position as computed from broadcast ephemerides given via the navigation file, and the postprocessed satellite positions for PRN14. When comparing with broadcast ephemerides we assume the precise positions to be the true ones.

The actual influence of the orbit error on the receiver position is given by the projection of the difference vector onto the line between the receiver and satellite. Let vector b be the difference vector between the satellite positions computed from the broadcast and the precise ephemerides, and a to be the vector between the receiver and the satellite position as computed from broadcast ephemerides. Then the projected difference vector p onto vector a is given as

$$p = \frac{a^T b}{a^T a} a.$$

Figure 3 shows vector p for PRN14. The computed ephemeris error varies in the range of ± 2 meters. The discontinuities in **EASY SUITE** continued on page 39

Success has a secret ingredient.

Whether they operate on land or sea or in the air, the companies who use SPAN™ are a quiet bunch. That's because NovAtel's tightly coupled GNSS/INS technology is an advantage they would rather keep to themselves. SPAN is currently inside many of the world's leading navigation applications, from in-air refuelling to autonomous navigation. And thanks to the new dual-antenna SPAN-SE-D, the ranks of successful silent types will grow considerably. To join them, visit novatel.com or call you know who.



Integrate success into your

EASY SUITE continued from page 25

the error reflect the routine ephemeris updates at two-hour intervals.

Additional Resources

Horn, R. A., and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, Cambridge, 1985


Author



Originally graduated as a chartered surveyor, **Kai Borre** obtained his Ph.D. in geodesy from Copenhagen University, and a Doctor of Technology degree from Graz University of Technology. He has been a full professor in geodesy at Aalborg University, Denmark, since 1976.

For more than 30 years Borre has conducted research and taught in the area of satellite-based positioning. In 1996 he established the Danish GPS Center and since 2000 has been head of a two-year international M.Sc. program in GPS technology.

Borre is a coauthor of the widely used textbooks, *Linear Algebra, Geodesy, and GPS* and *A Software-Defined GPS and Galileo Receiver; Single-Frequency Approach*.

Since the early 1990s, Borre has published Matlab code for processing of GPS observations. Since 2003 he has published Matlab code together with explanatory text, a successful new pedagogical concept. 

InsideGNSS
GPS | GALILEO | GLONASS | COMPASS