The GPS Easy Suite–Matlab code for the GPS newcomer

Kai Borre

Abstract The Matlab computing environment has become a popular way to perform complex matrix calculations, and to produce sophisticated graphics output in a relatively easy manner. Large collections of Matlab scripts are now available for a wide variety of applications and are often used for university courses. The GPS Easy Suite is a collection of ten Matlab scripts, or M-files, which can be used by those just beginning to learn about GPS. The first few scripts perform basic GPS calculations such as converting GPS Time in year/month/day/hour/minute/second format to GPS week/seconds of week, computing the position of a satellite using a broadcast ephemeris, and computing the coordinates of a single point using pseudorange observations. The latter scripts can perform calculations such as computing baseline components using either traditional least-squares or a Kalman filter, fixing cycle slips and millisecond clock jumps, and computing ionospheric delay using carrier phase observations. I describe the purpose of each M-file and give graphical results based on real data. The Matlab code and the sample datasets are available from my website. I have also included additional text files (in pdf format) to discuss the various Time Systems and Coordinate Systems used in GPS computations, and to show the equations used for computing the position of a satellite using the ephemeris information broadcast from the satellites.

Received: 8 January 2003 / Accepted: 24 January 2003
Published online: 4 April 2003

K. Borre
Danish GPS Center, Aalborg University,
Niels Jernes Vej 14, 9220 Aalborg, Denmark
E-mail: borre@gps.auc.dk
Tel.: +45-9635-8362
Fax: +45-9815-1583

Introduction

I seldom teach the same course the same way. Often I start afresh. Last autumn I tried a new approach to lecturing my Introductory GPS Computations class. Courses in my department run in modules of six lectures; so I split up the basic issues into six lectures containing the following topics:

– time: Universal Time, GPS Time (GPS week and seconds of week), Modified Julian Date, Leap Seconds, International Atomic Time, Terrestrial Dynamical Time (see time_itr.pdf on the web at http://www.gps.auc.dk/~borre/easy;
– Kepler's laws, computation of satellite position (see Strang and Borre 1997, pages 482–487);
– observation types, computation of receiver position (see Strang and Borre 1997, pages 463–465, 460–461);
– observational errors, differencing techniques, Dilution of Precision (DOP) (see Strang and Borre 1997, pages 453–460, 462–463, 465–467);
– computation of a baseline from pseudoranges;
– computation of a baseline from pseudoranges and phase observations (see Strang and Borre 1997, page 463).

Some reflections resulted in four additional topics:
– estimation of the receiver clock offset (see Strang and Borre 1997, pages 507–509);
– cycle-slip detection and repair of millisecond receiver clock resets (see Strang and Borre 1997, pages 491, 509);
– various representations of an estimated baseline (see Strang and Borre 1997, pages 367–368, 472–475, 501–502);
– ionospheric delay estimated from carrier phase observations Φ_1 and Φ_2 (see Strang and Borre 1997, page 490).

The basic GPS datasets we will be working with were collected at Aalborg by two JPS Eurocard receivers on 4 September 2001. The resulting Receiver Independent Exchange (RINEX) files (Gurtner 2000) are site247j.01o, site24~1.01o, and site247j.01n. For the more specialized topics we need a longer observation series; this is contained in file kofi1.01o.
We number the Matlab scripts easy1 to easy10; hence, they more or less relate to the ten topics listed above. All files are zipped and are available via the world wide web at http://www.gps.auc.dk/~borre/easy.

Much of the theoretical background for the easy scripts can be found in chapters 14 and 15 of Strang and Borre (1997). However, we wanted to add some additional text here to emphasize certain issues which are central to the code but often are not mentioned in textbooks. Below follows a description of the various M-scripts.

The Easy Suite

easy1

Nearly any GPS processing starts with time issues. Easy1 shows how to convert a given epoch, originating from one of the RINEX Observation files (O-files), to GPS week and seconds of week (sow). We include the time_itr.pdf file which is a text file which gives an overview of relevant timescales and reference frames useful for GPS.

easy2

The second basic computation is to calculate the position in the Earth-Centered, Earth-Fixed frame (ECEF) of a given PRN at a given time. For a given time and an ephemeris obtained from a RINEX Navigation Message file (N-file), easy2 does the job. The main function is satpos which is an implementation of the procedure described in the GPS Interface Control Document (ICD-GPS-200C 1997), Table 20-IV. We read a RINEX N-file and reformat it into Matlab's internal format, namely, a matrix named Eph. Furthermore, we relate each PRN to a column of Eph. Each column contains 21 variables; these comprise a complete ephemeris for one satellite.

easy3

In easy3 we compute a receiver's ECEF position from RINEX O- and N-files. Only pseudoranges are used. The computation is repeated over 20 epochs. Each position is the result of an iterative least-squares procedure. The variation of the position relative to the first epoch is shown in Fig. 1. The variation in coordinate values is typically less than 5 m. We open a RINEX O-file and read all the pseudoranges given at an epoch. With given satellite positions from easy2, we compute the receiver position.

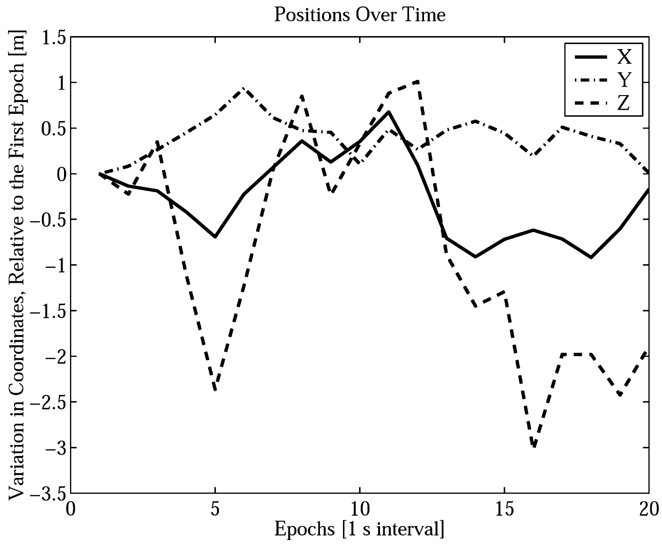

Fig. 1
Change in position over time using pseudoranges

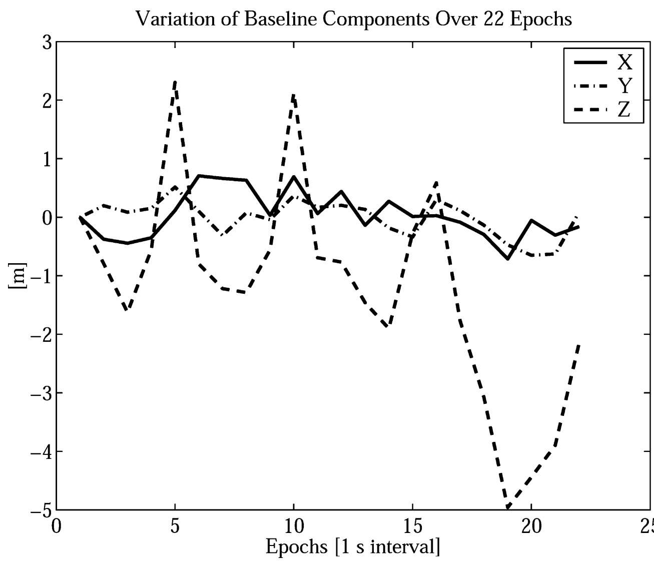

Fig. 2
Change in baseline components over time using pseudoranges

easy4

The easy4 file deals with simultaneous observations of C/A pseudoranges from two receivers. We estimate the baseline between the two antennas and plot the baseline components in Fig. 2. We obtain variations in the components of up to 10 m. So, an antenna point position and a baseline, both estimated from pseudoranges alone, do have the same noise level. This statement is valid only for observations taken after 2 May 2000.

easy5

So far all computations have been based on pseudoranges alone. Easy5 now includes phase observations as well. Apart from some more bookkeeping and the problem of ambiguity fixing, we do the same processing. For fixing the phase ambiguities we use the Lambda method; it has a Matlab implementation which we exploit through the call lambda2. With fixed ambiguities, we compute the baseline as a least-squares solution epoch by epoch. From Fig. 3 we immediately have the nice feature that the variation of the baseline components is now at the 10-mm level. That is, our results improve by a factor of 1,000 by including phase observations.

easy6

We now want to use a Kalman filter instead of a least-squares solution for the baseline estimation. We use an *extended Kalman filter* and we quote below the core lines of code. In filter terminology "extended" means nothing else but nonlinear.
Any Kalman filter needs initialization of three different covariance matrices: the covariance matrix P for the state

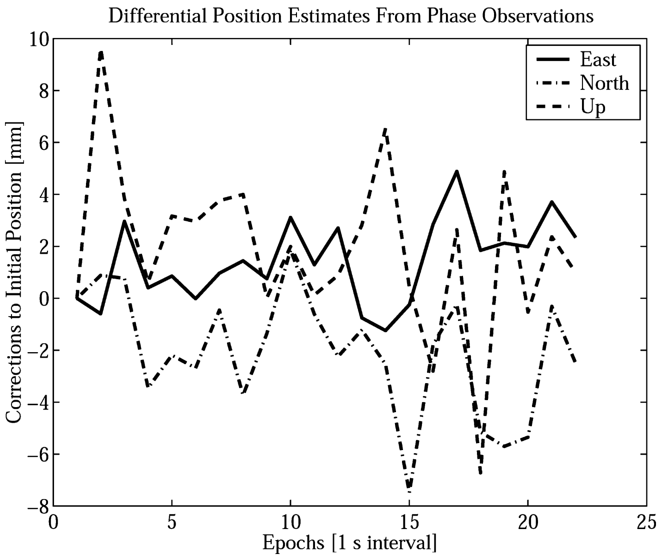

Fig. 3
Baseline components over time using pseudorange and phase observations

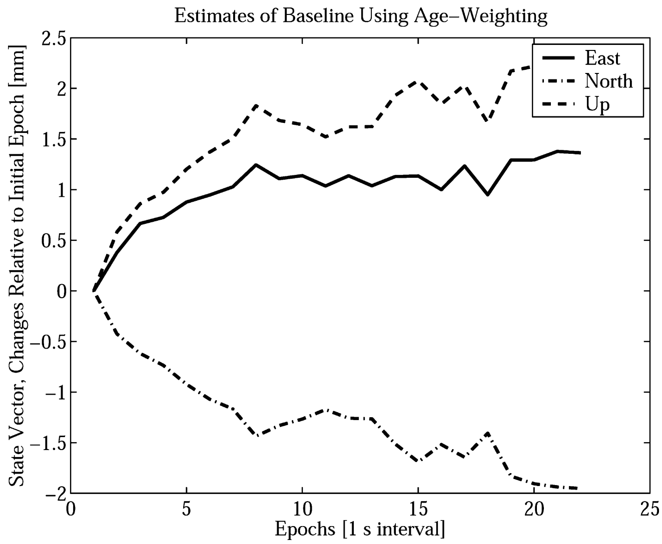

Fig. 4
Baseline estimated by an age-weighted, extended Kalman filter

vector x (the vector of unknowns), the covariance matrix Q for the system, and the covariance matrix R for the observations. A Sigma matrix takes care of the correlation of the observations introduced by using the double differencing technique.

- % The state vector contains (x,y,z)
- % Setting covariances for the Kalman filter
- P=eye(3); % covariances of state vector
- Q=0.05^2*eye(3); % covariances of system
- R=0.005^2*kron(eye(2),inv(Sigma)); % covariances of observations

The extended Kalman filter is implemented as four lines of code. Let the coefficient matrix of the linearized observation equations be A, and the right side is the observed minus computed values of the observations, $b-b_k$:

- $P=P+Q$;
- $K=P*A'*\mathrm{inv}(A*P*A'+R)$;
- $x=x+K*(b-b_k)$;
- $P=(\mathrm{eye}(3)-K*A)*P$.

The output for each epoch (update) is the state vector x (baseline) and the updated covariance matrix P for x. Note that there is no term like Ax because the innovation is $b-b_k$ rather than $b-Ax$.

easy6e

Easy6e is the same as easy6 except here the Kalman filter uses an overweighting of the most recent data. This method is also called *exponentially weighted recursive least squares* (Kailath et al. 2000).
We introduce a known scalar τ such that $0<\tau<1$. Since $\tau<1$, the factor τ^{i-j} weights past data (those which occur at time instants j which are sufficiently far from i) less heavily than more recent data (those which occur at time instants j relatively close to i). This feature enables an adaptive algorithm to respond to variations in data statistics by "forgetting" data from the remote past. Figure 4 shows the result using an age-weighted, extended Kalman filter. With the same data, but without age-weighting, you would get the plot shown in Fig. 3.
Note that the smaller τ is, the faster old observations are "forgotten". The introduction of the forgetting factor happened in the early 1960s. In retrospect, it is tremendous to realize that most of the tricks and computational options used in recent filter techniques were described within a few years after Kalman's landmark paper in 1960. We quote the core code of the algorithm:

- % the smaller tau is, the faster old observations are forgotten;
- tau=0.1;
- age_weight=exp(1/tau);
- % extended Kalman filter (see pages 509–510 in Strang and Borre (1997));
- $P=P+Q$;
- $K=P*A'*\mathrm{inv}(A*P*A'/\mathrm{age\_weight}+R)/\mathrm{age\_weight}$;
- $x=x+K*(b-b_k)$;
- $P=(\mathrm{eye}(3)-K*A)*P/\mathrm{age\_weight}$.

easy7

The "pseudo" part of the word pseudorange alludes to the receiver clock offset dt. Most often it is a parameter of less interest. However, in certain situations it is desirable to know dt. We supply an algorithm which delivers dt. In fact, it is an implementation of the algorithm given in example 15.1 in Strang and Borre (1997).
The actual data yield $dt \cong 0.38$ ms, as shown in Fig. 5. One can see that the receiver clock is fairly stable over a short period of time.

easy8

Any professional GPS software needs to check for cycle slips and resets of the receiver clock (Fig. 6). The reset

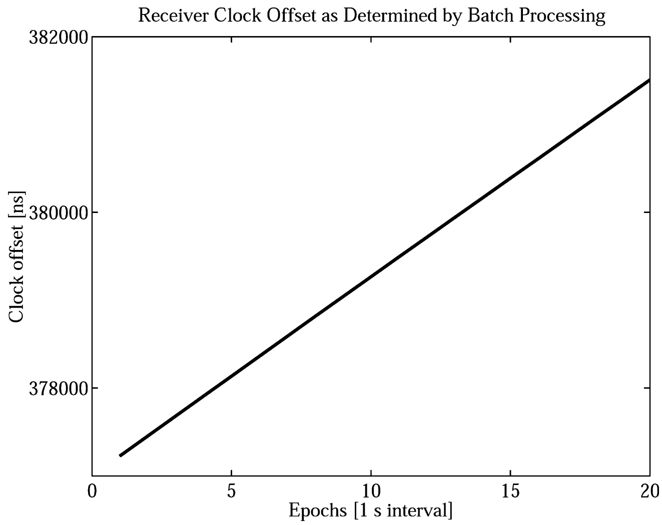

Fig. 5
Receiver clock offset dt

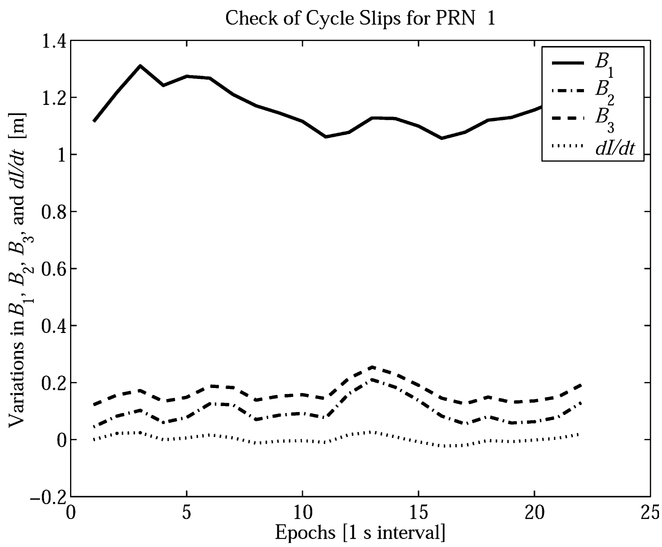

Fig. 6
Check of cycle slips

amount of receiver clock is typically one millisecond, and it influences the pseudoranges alone, while cycle slips spoil the carrier phase observations.

The preliminary data validation can be based on the single differenced observations (i.e., differences between two receivers). The goal is to detect cycle slips and outliers in the GPS single difference observations without using any external information with regard to satellite and receiver dynamics, their clock behavior and atmospheric effects. It is done independently for each satellite. There is therefore no minimum number of satellites required for this so-called *integrity monitoring* to work. The following is based on an idea by Kees de Jong (de Jong 1998).

The dual-frequency single difference measurement model cannot be used directly as it is, since this model is singular. In order to make it regular, a re-parameterization has to be performed. Let $\alpha=(f_1/f_2)^2$, and let hardware delays be denoted η, then the measurement model for epoch k reads

$$\begin{bmatrix} P_1 \\ P_2 \\ \Phi_1 \\ \Phi_2 \end{bmatrix}_k = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} [\rho + c \cdot dt + T + I + \eta_{P1}]_k + \begin{bmatrix} 0 \\ \eta_{P2} - \eta_{P1} + (\alpha - 1) I_k \\ \eta_{\Phi1} - \eta_{P1} - 2I_k + \lambda_1 N_1 \\ \eta_{\Phi2} - \eta_{P1} + (-\alpha - 1) I_k + \lambda_2 N_2 \end{bmatrix} \quad (1)$$

where P_1 and P_2 are the pseudorange observables, Φ_1 and Φ_2 are the carrier phase observables, N_1 and N_2 are the carrier phase ambiguities, λ_1 and λ_2 are the wavelengths associated with frequencies f_1 and f_2, ρ is the geometric range, dt is the receiver clock oset, T is the tropospheric delay, and I is the ionospheric delay (all terms are single-dierenced quantities in units of meters). In abbreviated notation this gives

$$\begin{bmatrix} P_1 \\ P_2 \\ \Phi_1 \\ \Phi_2 \end{bmatrix}_k = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} R_k + \begin{bmatrix} 0 & 0 & 0 \\ \alpha - 1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -\alpha - 1 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix}_k \quad (2)$$

with covariance matrix $\Sigma_b=2\Sigma$ and

$$\Sigma = \begin{bmatrix} \sigma_{P1}^2 & 0 & 0 & 0 \\ 0 & \sigma_{P2}^2 & 0 & 0 \\ 0 & 0 & \sigma_{\Phi1}^2 & 0 \\ 0 & 0 & 0 & \sigma_{\Phi2}^2 \end{bmatrix} \quad (3)$$

$$R_k = [\rho + c \cdot dt + T + I + \eta_{P1}]_k \quad (4)$$

$$B_1 = I_k - \frac{\eta_{P1} - \eta_{P2}}{\alpha - 1} \quad (5)$$

$$B_2 = I_k + \frac{\eta_{P1} - \eta_{\Phi1}}{2} - \frac{\lambda_1 N_1}{2} \quad (6)$$

$$B_3 = I_k + \frac{\eta_{P1} - \eta_{\Phi2}}{\alpha + 1} - \frac{\lambda_2 N_2}{\alpha + 1} \quad (7)$$

Parameter R_k in general does not change smoothly with time and is therefore hard to model using, for example, low-degree polynomials. It will therefore be eliminated by pre-multiplying the left and right sides of the above measurement model by the transformation matrix M, defined as

$$M = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

resulting in

$$\begin{bmatrix} P_2 - P_1 \\ \Phi_1 - P_1 \\ \Phi_2 - P_1 \end{bmatrix}_k = \begin{bmatrix} \alpha - 1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -\alpha - 1 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix}_k \quad (9)$$

with covariance matrix $M\,\Sigma_b\,M^T$. The parameters B_1, B_2, and B_3 are linear combinations of the time-dependent ionospheric eect and the constant hardware delays and carrier ambiguities. The ionospheric eect will be modeled as a first-order polynomial, i.e., as a bias I_k and a drift $\dot{I}_k$. The dynamic model reads

$$\begin{bmatrix} I \\ \dot{I} \end{bmatrix}_k = \begin{bmatrix} 1 & t_k - t_{k-1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I \\ \dot{I} \end{bmatrix}_{k-1} \tag{10}$$

The dynamic model for all parameters then becomes

$$\begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ \dot{I} \end{bmatrix}_k = \begin{bmatrix} 1 & 0 & 0 & t_k - t_{k-1} \\ 0 & 1 & 0 & t_k - t_{k-1} \\ 0 & 0 & 1 & t_k - t_{k-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ \dot{I} \end{bmatrix}_{k-1} \tag{11}$$

and the measurement model becomes

$$\begin{bmatrix} P_2 - P_1 \\ \Phi_1 - P_1 \\ \Phi_2 - P_1 \end{bmatrix}_k = \begin{bmatrix} \alpha - 1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & -\alpha - 1 & 0 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ \dot{I} \end{bmatrix}_k . \tag{12}$$

With the above measurement model and dynamic model, it is possible to detect cycle slips as small as one cycle in the carrier observations, without using any external information, even for relatively large observation intervals (or data gaps) t_k–t_{k-1}.

easy9

This script converts the estimated baseline to topocentric geodetic azimuth, elevation angle and slant distance, and applies covariance propagation. The input of the covariance propagation is the 3×3 covariance matrix obtained from the baseline estimation.

easy10

If we use a dual-frequency receiver we can estimate the ionospheric delay I_k at epoch k as

$$I_k = \alpha\left(\Phi_{2,k} - \Phi_{1,k}\right) - \alpha(\lambda_2 N_2 - \lambda_1 N_1). \tag{13}$$

We are only interested in changes of I_k over time for the individual PRNs, so we may omit the last, constant term. This leaves the simple expression for the ionospheric delay

$$I_k = \alpha\left(\Phi_{2,k} - \Phi_{1,k}\right). \tag{14}$$

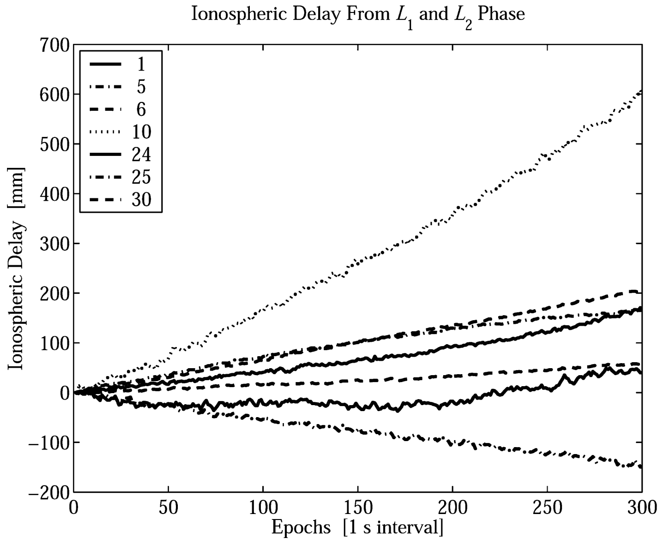

Fig. 7
Ionospheric delay as a function of time

Estimates of I_k for the various PRNs are shown in Fig. 7. The data are from file kofi1.01o.

References

de Jong K (1998) Real-time integrity monitoring, ambiguity resolution and kinematic positioning with GPS. In: Proc 2nd European Symp GNSS'98, Toulouse, pp VIII07/1–VIII07/7

Gurtner W (2000) RINEX: the receiver independent exchange format version 2.10. Available from ftp://igscb.jpl.nasa.gov/igscb/data/format/rinex210.txt

ICD-GPS-200C (1997) Interface control document, IRN-200C-002, 25 September 1997. ARINC Research Corporation, Fountain Valley, CA

Kailath T, Sayed A, Hassibi B (2000) Linear estimation. Prentice Hall, Englewood Cliffs, NJ

Strang G, Borre K (1997) Linear algebra, geodesy, and GPS. Wellesley-Cambridge Press, Wellesley, MA. Order at www.wellesleycambridge.com