Autocorrelation Functions in GPS Data Processing: Modeling Aspects

Kai Borre, *Aalborg University*

Gilbert Strang, *Massachusetts Institute of Technology*

Helsinki University of Technology   November 12, 2001

Consider a process that is actually random walk but is *incorrectly* modeled as a random constant c. We have then for the true model

$$x_k = x_{k-1} + \epsilon_k, \qquad \epsilon_k = \text{unity Gaussian white noise},$$

$$\sigma^2\{x_0\} = 1$$

$$b_k = x_k + e_k, \qquad k = 0, 1, 2, \ldots \qquad \sigma^2\{e_k\} = 0.1$$

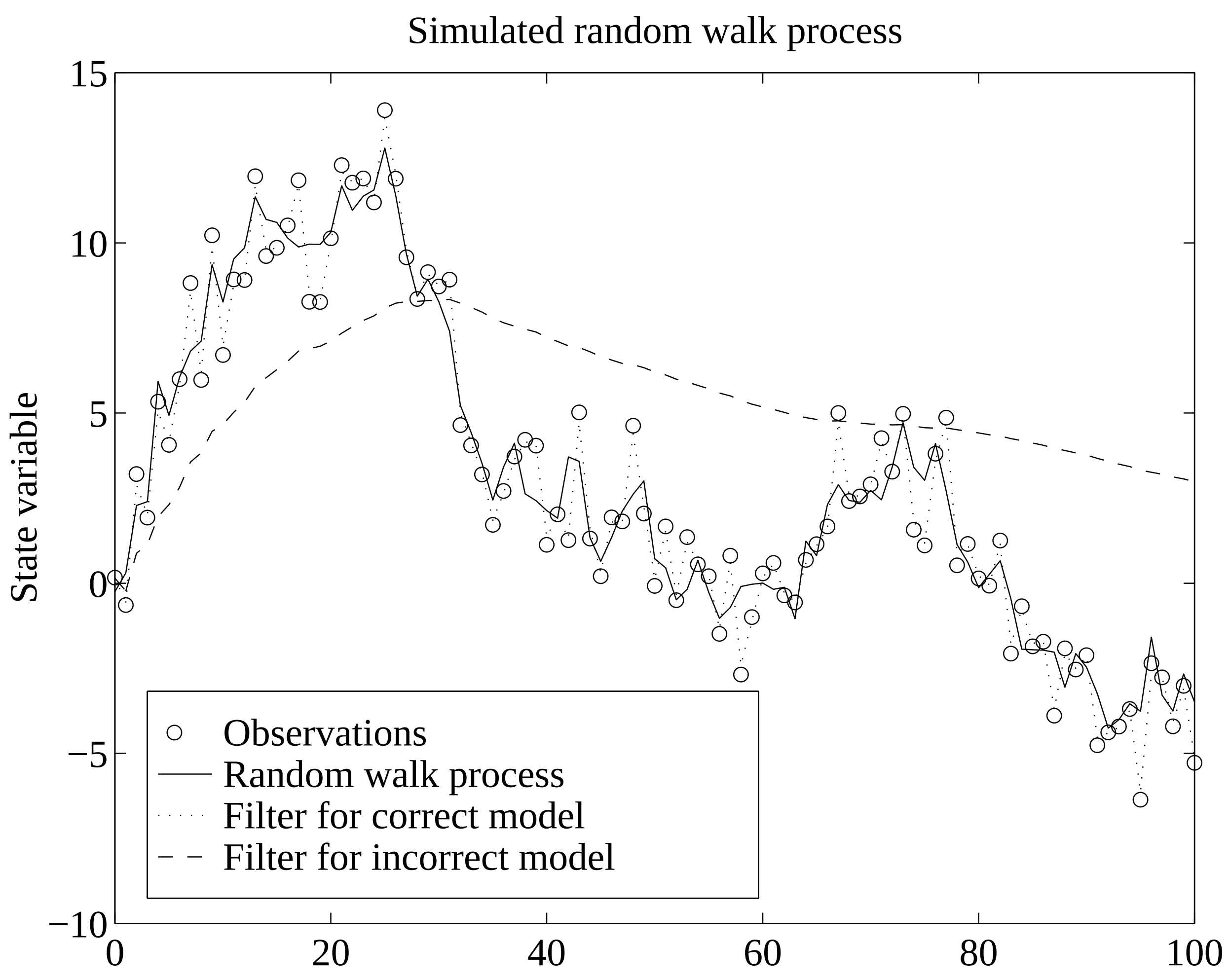
and for the incorrect model

$$x_k = c, \qquad c \sim N(0,1)$$

$$b_k = x_k + e_k, \qquad k = 0, 1, 2, \ldots \qquad \sigma^2\{e_k\} = 0.1$$

The wrong model has $F_k = 1$, $\Sigma_{\epsilon,k} = 0$, $A_k = 1$, $\Sigma_{e,k} = 0.1$, $\hat{x}_0 = 0$, and $P_0^{-1} = 1$. For the true model the parameters are the same except that $\Sigma_{\epsilon,k} = 1$, rather than zero.

Correctly and incorrectly filtered random walk process



Computing the Autocorrelation Function

$$\text{Shift} = 0: \quad \begin{array}{cccccc} a_0 & a_1 & a_2 & a_3 & a_4 & \ldots \\ a_0 & a_1 & a_2 & a_3 & a_4 & \ldots \end{array}$$

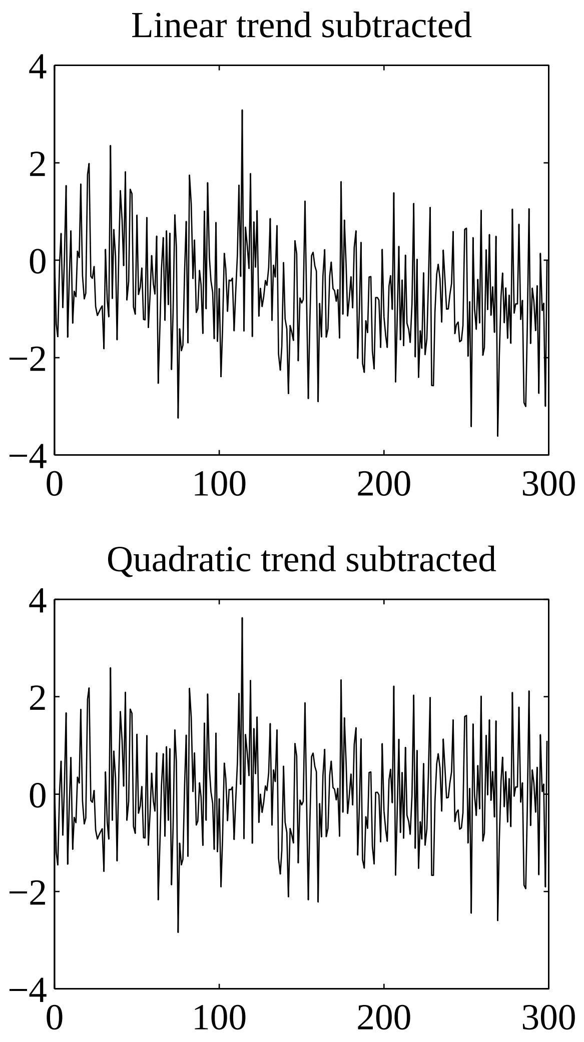
$$\text{auto}(0) = \sum_0^{n-1} a_i a_i / n.$$

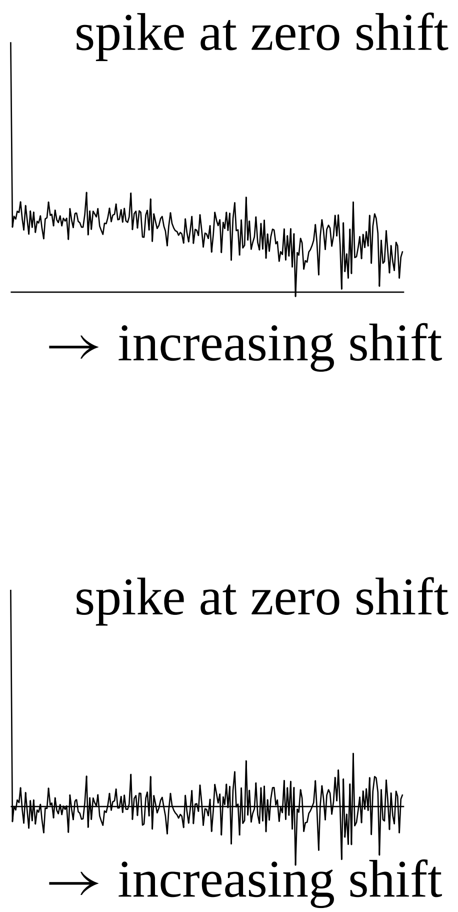
We multiply elements $a_i a_i$ above each other, and add. Now shift the lower row:

$$\text{Shift} = 1: \quad \begin{array}{ccccccc} a_0 & a_1 & a_2 & a_3 & a_4 & \ldots & \\ & a_0 & a_1 & a_2 & a_3 & a_4 & \ldots \end{array}$$

$$\text{auto}(1) = \sum_1^{n-1} a_i a_{i-1} / n.$$

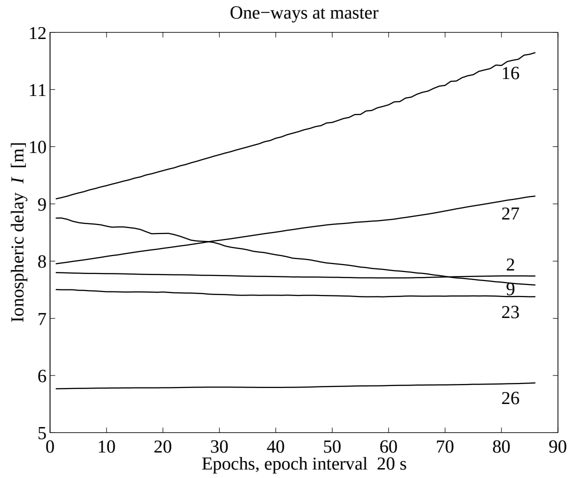
Autocorrelation for random data. The peak at zero measures the variance

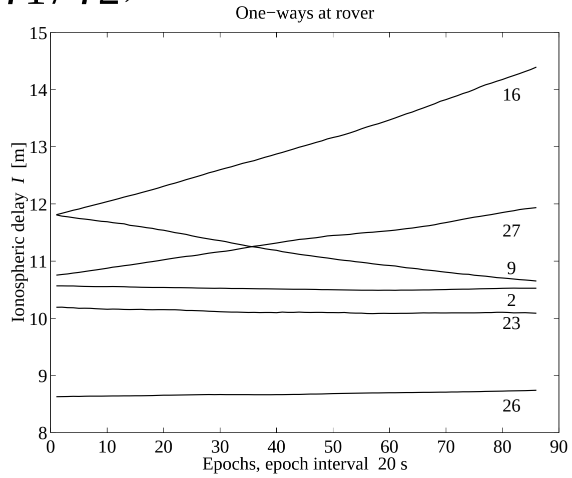


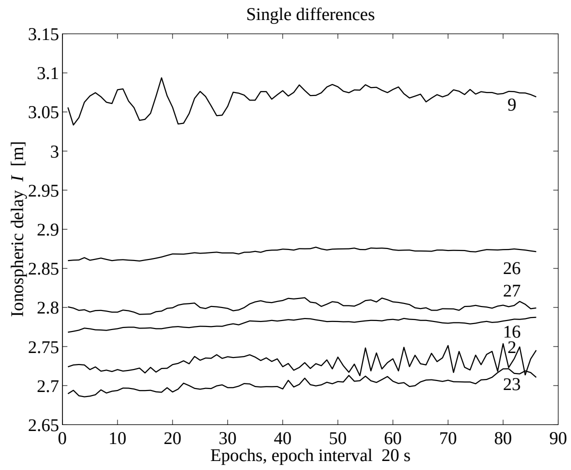


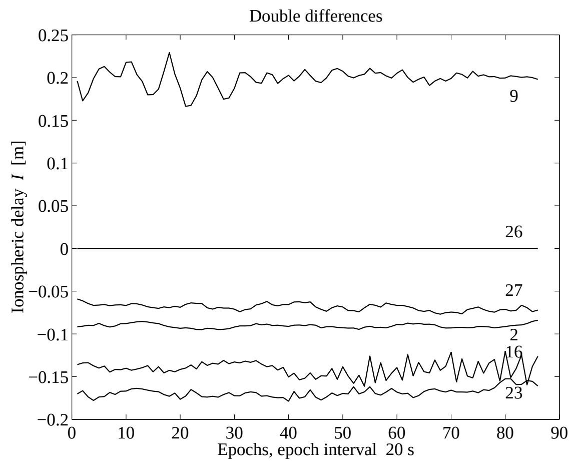
Ionospheric delays for one-ways, single and double differences

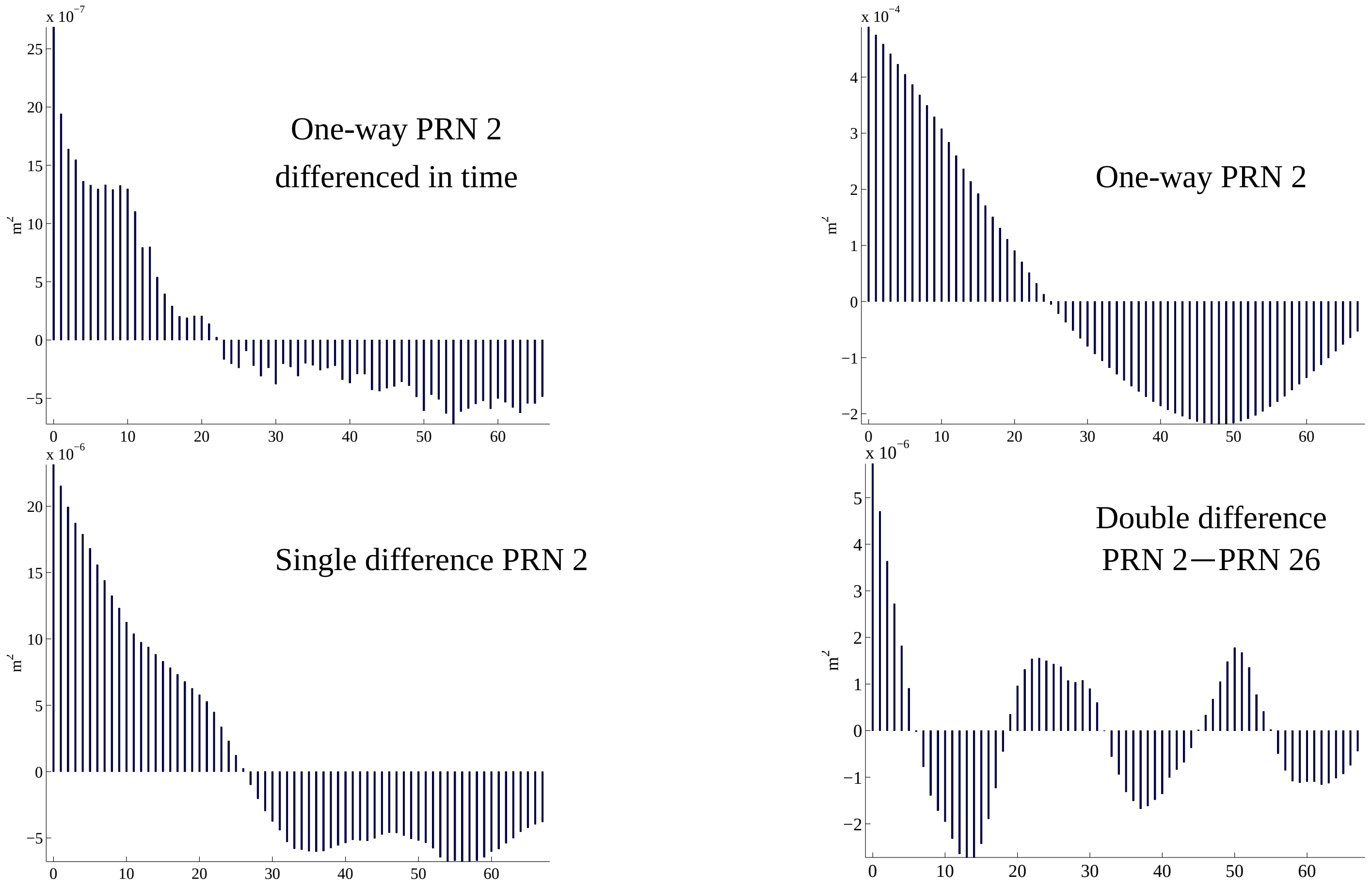
$$I_k = \frac{(\Phi_{2,k} - \lambda_2 N_2) - (\Phi_{1,k} - \lambda_1 N_1)}{1 - (f_1/f_2)^2}$$











Autocorrelation for ionospheric delay to PRN 2. Upper left shows *difference in time*: $I_k - I_{k-1}$, upper right the one-way, lower left the single difference, and lower left the double difference. Note the various orders of magnitude

Autocorrelation of ionosphere delay for one-ways

PRN	Elevation (in °)	σ_I (in m) master	σ_I (in m) rover	Shift for first zero master	Shift for first zero rover
26	68.9	0.08	0.11	35	30
2	59.0	0.08	0.04	15	35
27	28.0	0.39	0.35	30	32
16	22.8	0.77	0.71	30	30
23	20.4	0.17	0.19	12	20
9	18.5	0.48	0.19	15	30

Reference

Borre, Kai & Gilbert Strang (1997) *Autocorrelation Functions in GPS Data Processing: Modeling Aspects*. Pages 1143–1150. Institute of Navigation GPS-97, September 16–19, Kansas City, Missouri

Kalman Filters, Correlation Functions, Power Spectrums, and Ambiguity Experiments

Kai Borre, *Aalborg University*

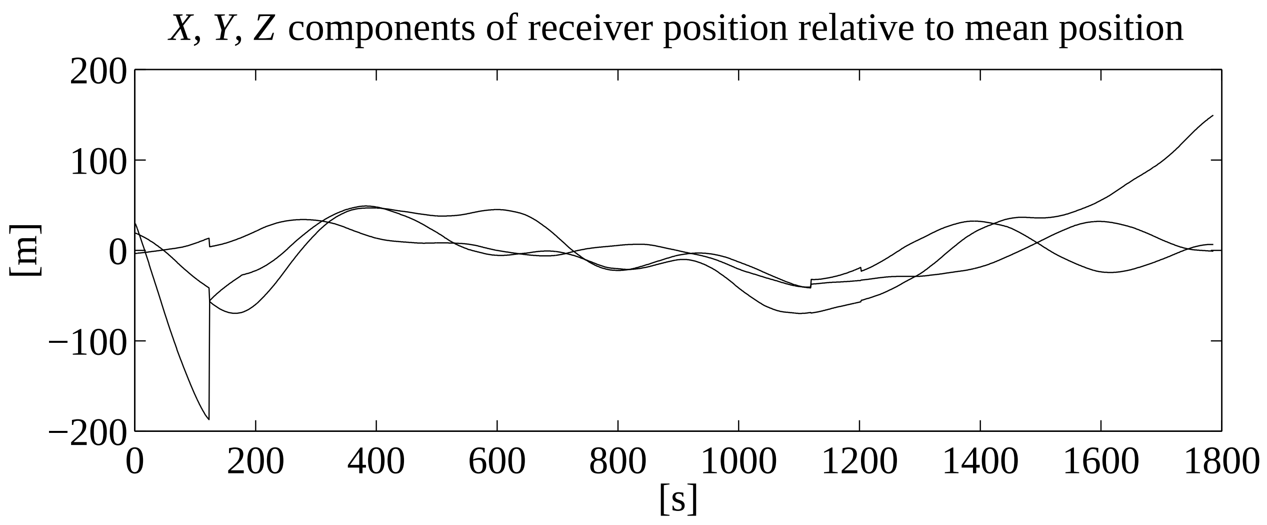
Helsinki University of Technology November 12, 2001

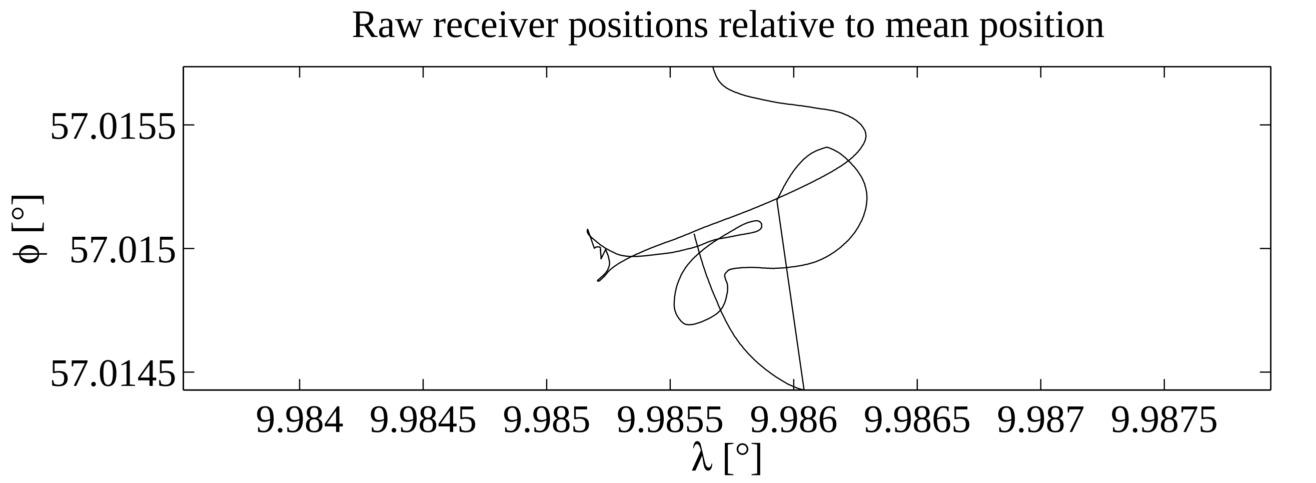
The following matrix equation

$$\begin{bmatrix} P_1 \\ \Phi_1 \\ P_2 \\ \Phi_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & \lambda_1 & 0 \\ 1 & (f_1/f_2)^2 & 0 & 0 \\ 1 & -(f_1/f_2)^2 & 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \rho \\ I \\ N_1 \\ N_2 \end{bmatrix} - \begin{bmatrix} e_1 \\ \epsilon_1 \\ e_2 \\ \epsilon_2 \end{bmatrix}. \tag{1}$$

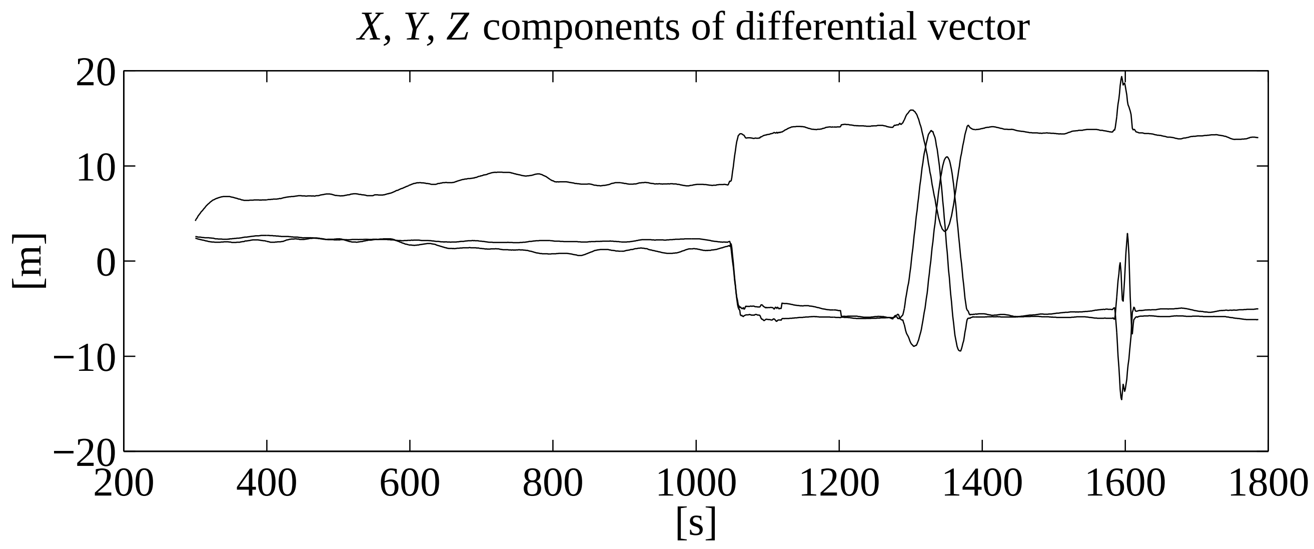
can be used in various ways. One method is to solve for the 4 unknowns on an epoch-to-epoch basis. An alternative is to employ (1) as an observation equation in a filter. In any case we are faced with an interesting problem: ρ and I are to be estimated as reals while N_1 and N_2 by definition are *integers*. This situation is difficult. One procedure is to solve for all 4 unknowns as were they reals. This gives what is called a ***float solution***. Given the float solution we can use a smart method that estimates N_1 and N_2 as integers. This refined solution is known as the ***fixed solution***. A popular method is the LAMBDA method decscribed by Teunissen [4]. However, our Matlab code uses a less complicated method described in [5].

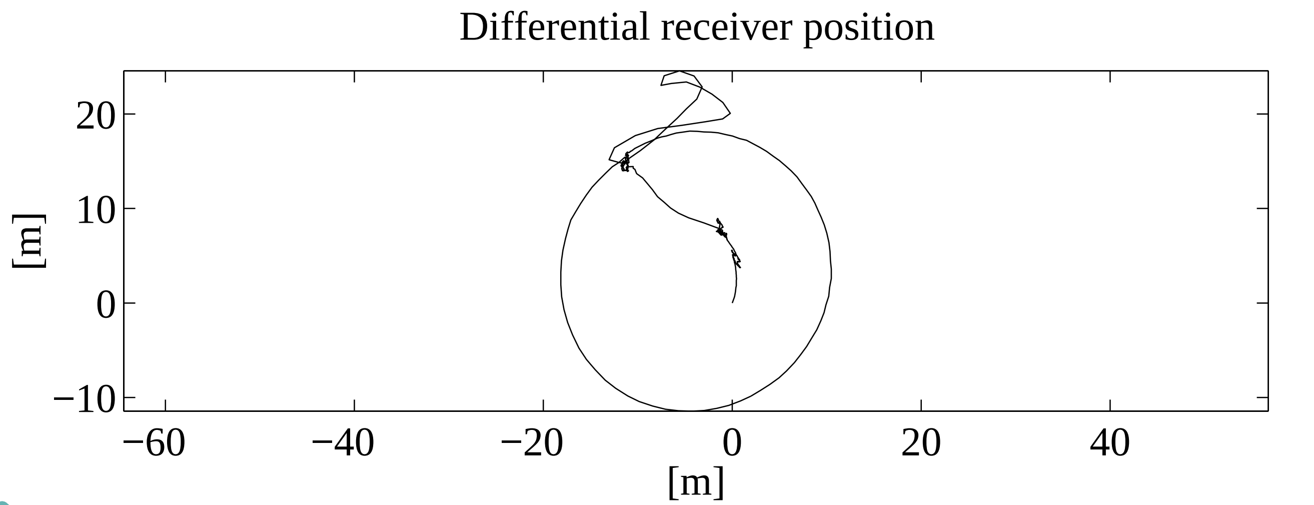
Computed positions for a static receiver. The present observations are taken with active SA. The variation over time is mainly due to SA



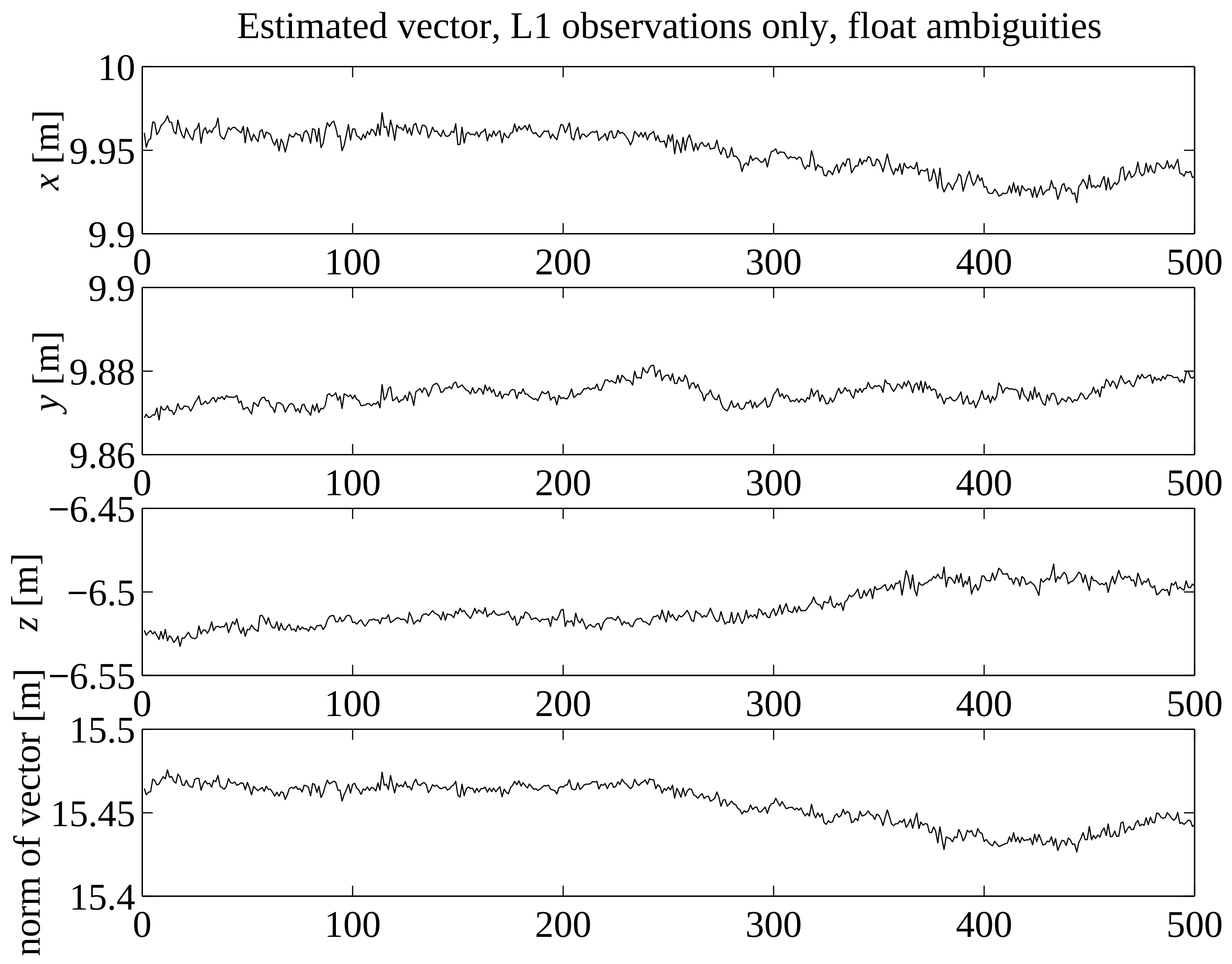


The first 300 epochs are deleted due to bad data. During epochs 300–1050 the rover remains at rest, from epoch 1050 it is moving a bit. During epochs 1291–1380 it is moved in a circle with radius ca. 16 meters. Then it rests at the circumference in epochs 1380–1587 and then makes a random movement outside the circle in epochs 1615–1787. The difference in computed positions for a static (master) receiver and a moving (rover) receiver. Now the detailed picture of the actual movement becomes clear. The ***differential*** positions are good to within a few dm.

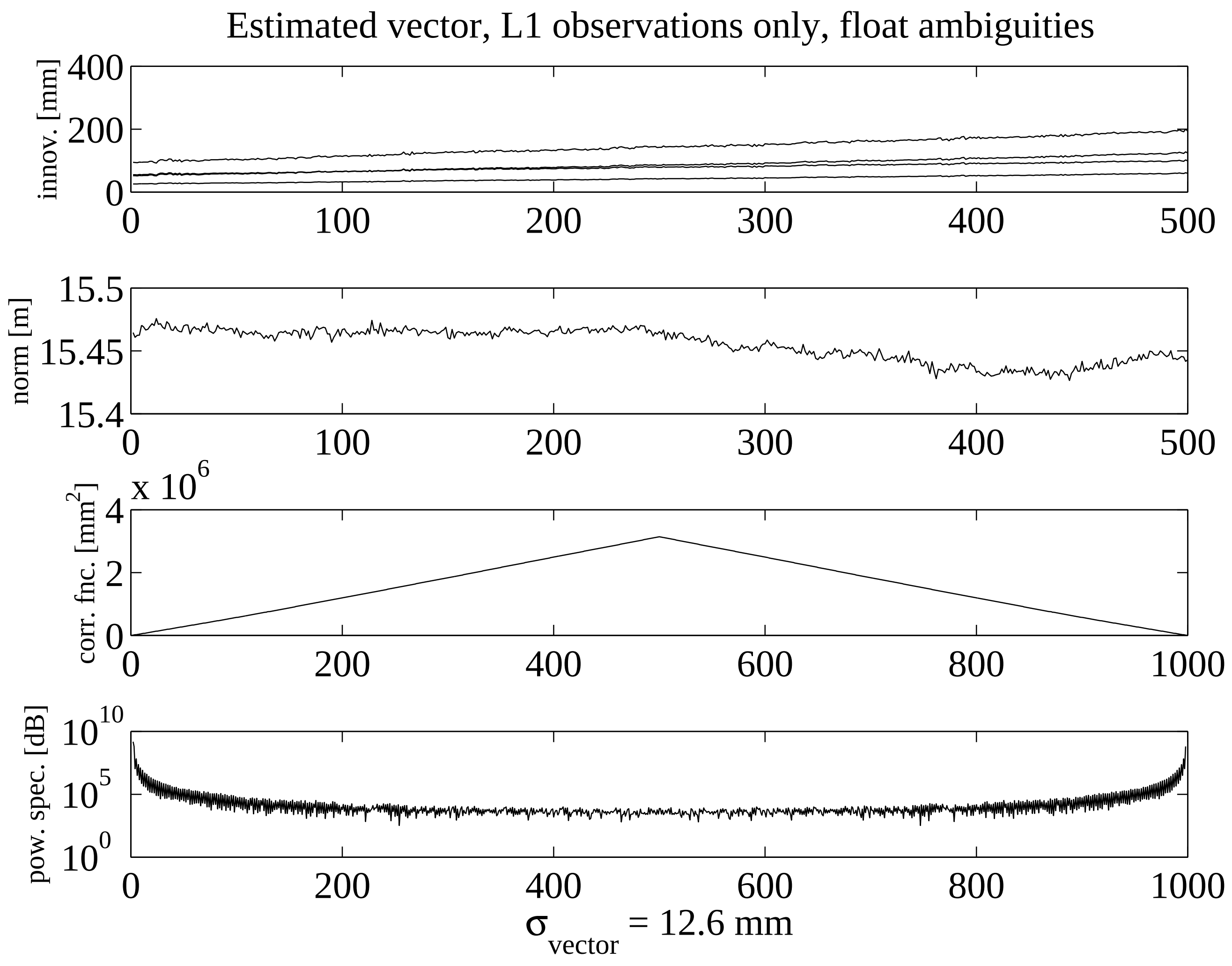




Double differenced code and phase observations on L_1 from static receivers. The ambiguities N_1 are estimated as a *float solution* based on observations from epochs 300–400. Next the vector components (x, y, z) are computed and the vector length.



The figure shows the innovation for the four satellites as computed with respect to the reference satellite, covariance function for a selected innovation component, and the corresponding power spectrum illustrating the color of the innovation noise.



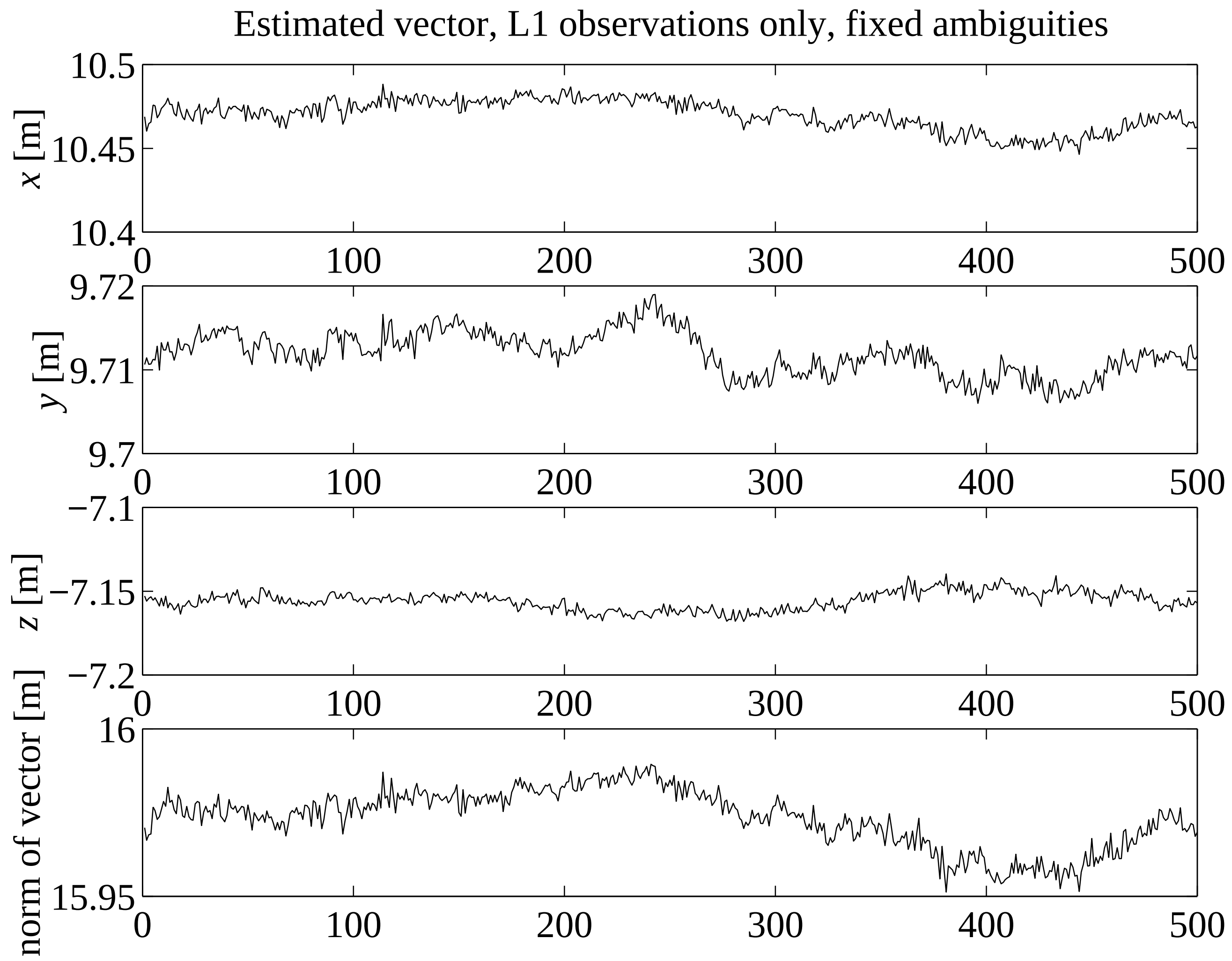
Float and fixed values of L_1 ambiguities:

−221 971.00	−221 970
664 919.67	664 918
163 531.36	163 531
1 198 426.78	1 198 429

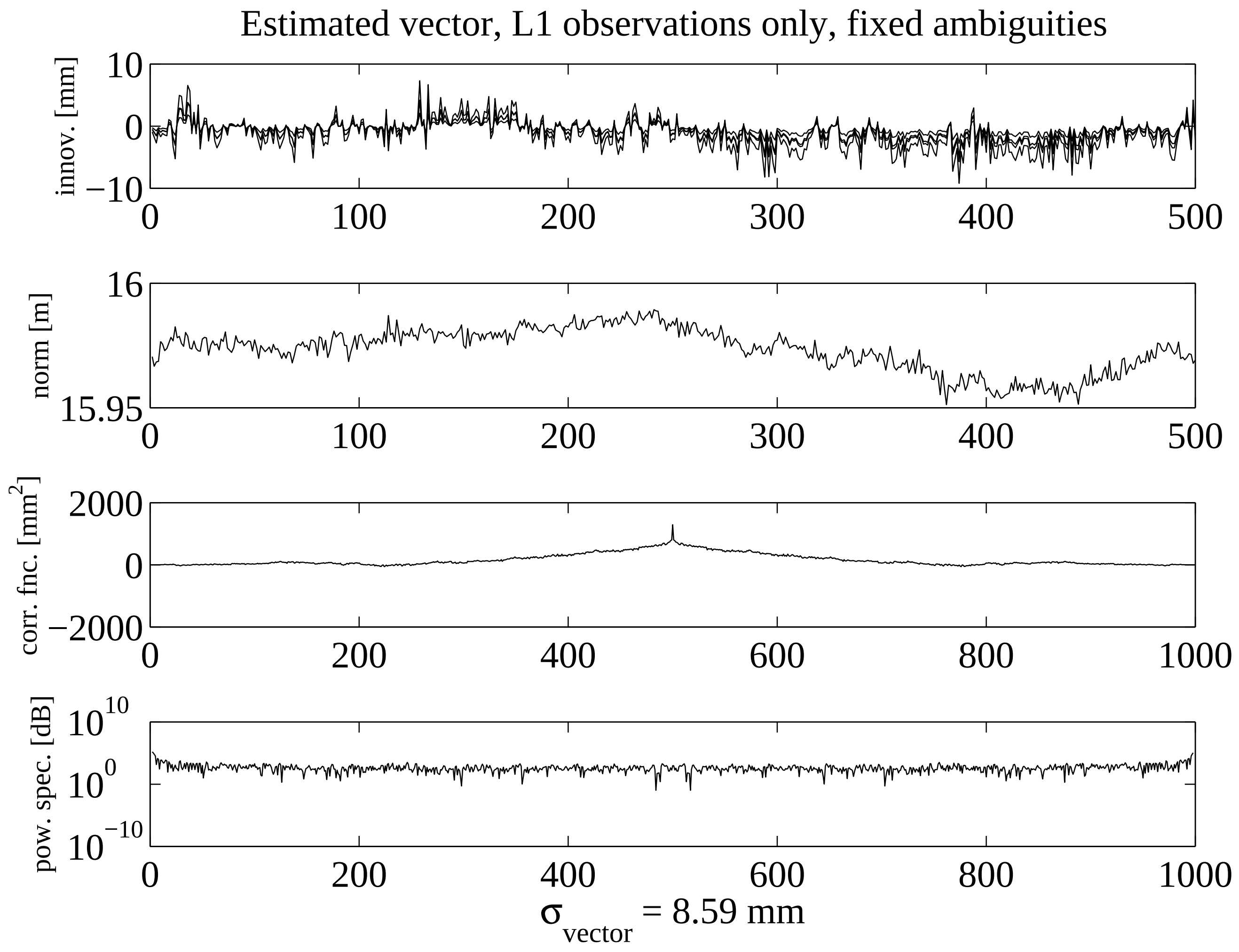
The fixed values result from a procedure given by Yang et al. [5]. The above table lists the estimated ambiguities. We observe that the rounded values of the float ambiguities deviate at most 2 from the fixed values.

The standard deviation of the length of the vector between master and rover is $\sigma_{\text{vector}} = 16.7\,\text{mm}$ for the float solution and $\sigma_{\text{vector}} = 8.3\,\text{mm}$ for the fixed solution. We see that the vector components drift over time when incorrect ambiguities are used. Fixing the ambiguities to the correct values yields an estimate of the vector components at cm-level or even at mm-level.

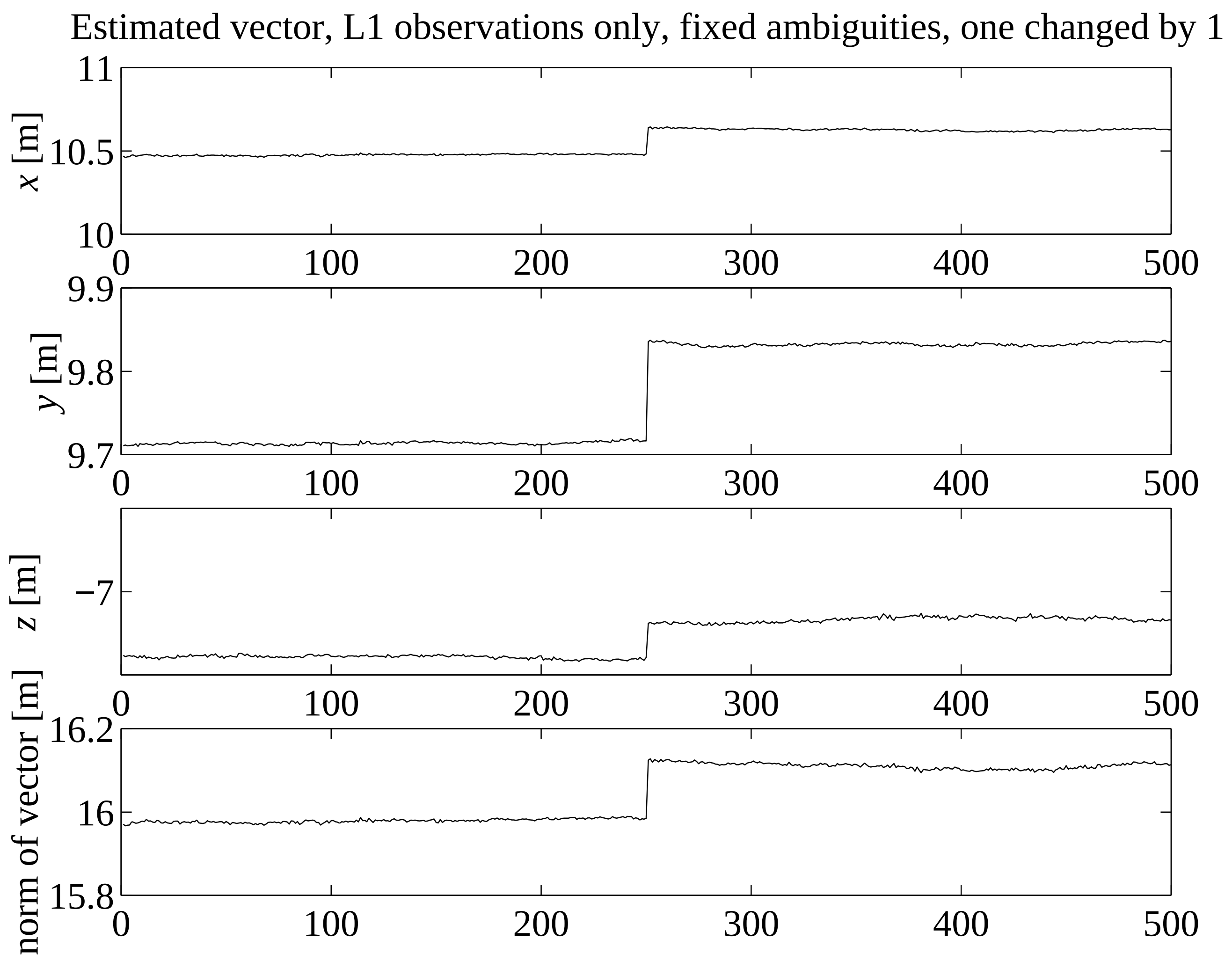
Double differenced code and phase observations on L_1 from static receivers. Note that the vector length is easier to estimate than its single components x, y, z.



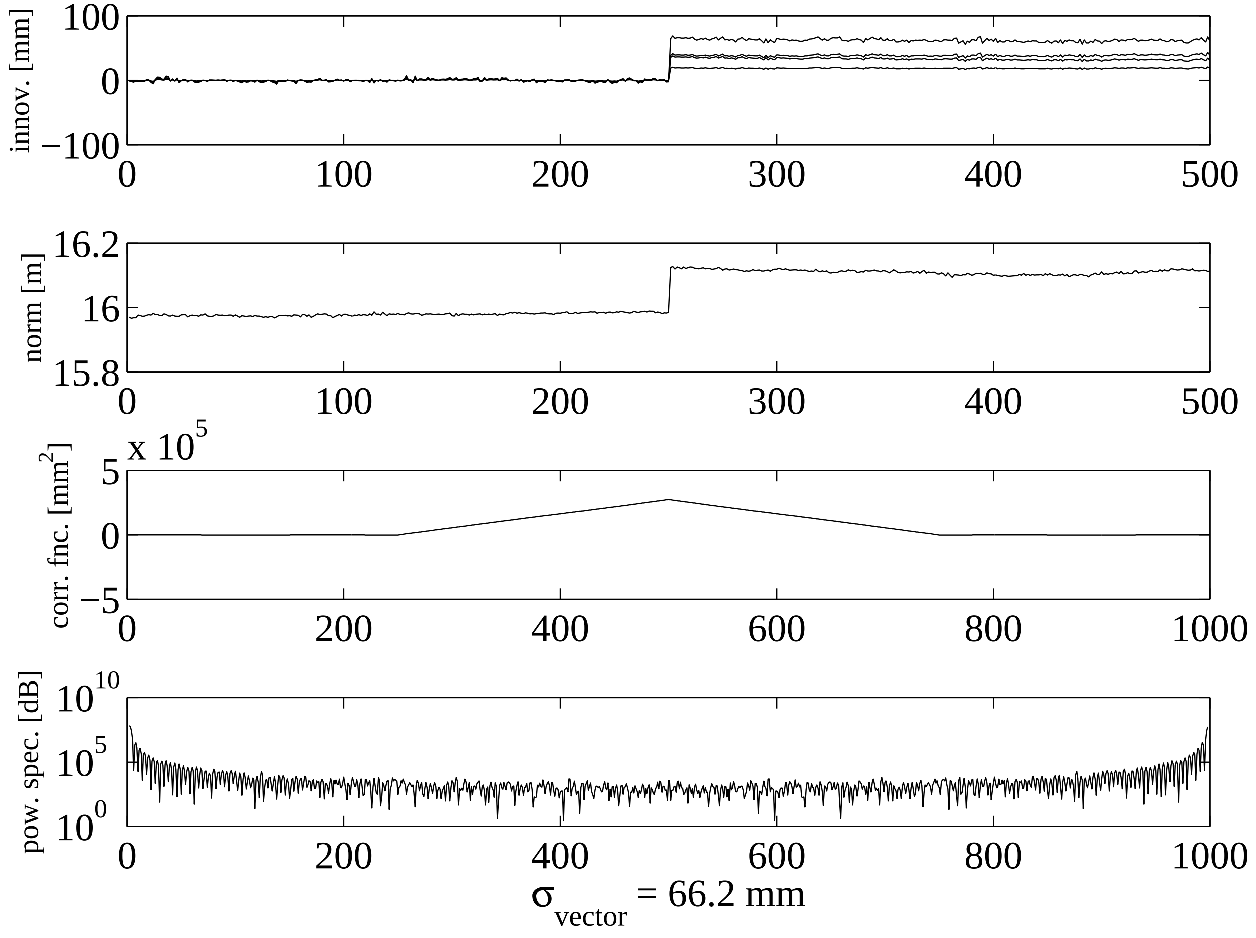
The ambiguities N_1 are estimated as a ***fixed solution*** based on observations from epochs 300–400. Note the dramatic increase in accuracy by fixing the ambiguities! Note the nice peak of the covariance function. The innovations are close to white noise.



The same as Figure 3, but at epoch 250 we ***change an ambiguity*** with the amount of 1 cycle



Estimated vector, L1 observations only, fixed ambiguities, one changed by 1 cycle



The length of the baseline vector $\|\boldsymbol{v}\| = \sqrt{x^2 + y^2 + z^2} = d$ often is easier to estimate than the components of $\boldsymbol{v} = (x, y, z)$. We illustrate this by applying the law of variance propagation:

$$\sigma_d^2 = \begin{bmatrix} \frac{\partial d}{\partial x} & \frac{\partial d}{\partial y} & \frac{\partial d}{\partial z} \end{bmatrix} \Sigma_v \begin{bmatrix} \frac{\partial d}{\partial x} \\ \frac{\partial d}{\partial y} \\ \frac{\partial d}{\partial z} \end{bmatrix} = \left(\frac{1}{d}\boldsymbol{v}^{\mathrm{T}}\right)\Sigma_v\left(\frac{1}{d}\boldsymbol{v}\right).$$

For the actual data and with fixed ambiguities we have $\sigma_d = 3.5\,\text{mm}$, $\sigma_x = 4.8\,\text{mm}$, $\sigma_y = 1.7\,\text{mm}$, and $\sigma_z = 3.2\,\text{mm}$. This partially supports our statement.

All M-files producing the figures in this paper can be found at
kom.auc.dk/~borre/life-l99

The original Ashtech data are stored in files ***.105**. The M-files **bdata**, **edata**, and **sdata** read the binary observation files and rearrange the data into a useful MATLAB format in the ***.dat** files.

The call **rawrecpo** produces our first figure. This file calls two other M-files **togeod** and **topocent**. **togeod** is a function that computes geodetic coordinates (ϕ, λ, h) given Cartesian coordinates (X, Y, Z) and the semi-major axis a and inverse flattening f for a reference ellipsoid. **topocent** transforms a 3-D vector dx into topocentric coordinates $(Dist, Az, El)$ with origin at X.

The main M-file **makebase** reads the ***.dat** files, estimates ambiguities for DD on L1 according to Yang et al. (1994) and uses **get_eph** and **find_eph** to compute satellite positions. The file **get_rho** computes the distance in ECEF between satellite and receiver at raw transmit time given an ephemeris. Now we can write the observation equations for the unknown vector components (x, y, z). We solve the normals on an epoch-to-epoch basis.

Conclusions

We have given a standard presentation of GPS observations and their characteristics. With a real data set we focus on a few specific topics: float and fixed ambiguity solutions and the effect on the estimated vector. In our special case we note that the standard deviation of the vector length is halved when changing from float to fixed solution. We also note that it is easier to estimate the vector length rather than the vector components themselves. Under controlled circumstances we change an ambiguity and study the effect on the solution and observe how the innovation vector slowly strives for the correct known solution.

References

[1] Borre, Kai (1995) *GPS i Landmålingen*. Order at **borre@kom.auc.dk**

[2] Borre, Kai (2001) *The GPS Toolbox*. GPS Solutions, **4:3**:47–51. John Wiley & Sons, Inc.

[3] Strang, Gilbert & Borre, Kai (1997) *Linear Algebra, Geodesy, and GPS*. Wellesley-Cambridge Press. Order at **www.wellesleycambridge.com**

[4] Teunissen, P. J. G. (1993) *Least-Squares Estimation of the Integer GPS Ambiguities*. In: LGR-Series No. 6, Delft Geodetic Computing Center

[5] Yang, Ming & Goad, Clyde & Schaffrin, Burkhard (1994) *Real-time On-the-Fly Ambiguity Resolution Over Short Baselines in the Presence of Anti-Spoofing*. In Proceedings of ION GPS-94, 519–525

Block Elimination and Weight Matrices

Kai Borre

Aalborg University

Helsinki University of Technology   November 12, 2001

Block Elimination

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix}.$$

$$\begin{bmatrix} I & 0 \\ -CA^{-1} & I \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} = \begin{bmatrix} I & 0 \\ -CA^{-1} & I \end{bmatrix} \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix}$$

or explicitly

$$\begin{bmatrix} A & B \\ 0 & D - CA^{-1}B \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 - CA^{-1}\boldsymbol{b}_1 \end{bmatrix}.$$

Final result

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1}(I + B(D - CA^{-1}B)^{-1}CA^{-1}) & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}.$$

Schreiber's Device

In 1877 Schreiber described a procedure for eliminating the unknowns $\boldsymbol{x}_1$ (we present a formulation somewhat more general than the original one):

> Let be given a set of observation equations $A\begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} = \boldsymbol{b} - \boldsymbol{\epsilon}$. The unknown $\boldsymbol{x}_1$ is eliminated if we delete the columns in A corresponding to the unknown $\boldsymbol{x}_1$, and add the fictitious observational row
>
> $$\begin{bmatrix} \boldsymbol{a}_1^{\mathrm{T}}\Sigma^{-1}\boldsymbol{a}_2 & \boldsymbol{a}_1^{\mathrm{T}}\Sigma^{-1}\boldsymbol{a}_3 & \dots & \boldsymbol{a}_1^{\mathrm{T}}\Sigma^{-1}\boldsymbol{a}_n \end{bmatrix}$$
>
> having the negative weight $-1/\boldsymbol{a}_1^{\mathrm{T}}\Sigma^{-1}\boldsymbol{a}_1$.

We demonstrate the procedure on a numerical example:

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 3 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 8 \\ 8 \\ 20 \end{bmatrix}.$$

The augmented left side of the normals $A^{\mathrm{T}}\Sigma^{-1}A$ is

$$\begin{bmatrix} 0 & 1 & 3 & 4 & 8 \end{bmatrix} \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & -\frac{1}{4} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 3 \\ 4 \\ 8 \end{bmatrix} = \begin{bmatrix} 10 \end{bmatrix}.$$

The augmented right side of the normals $A^{\mathrm{T}}\Sigma^{-1}\boldsymbol{b}$ is

$$\begin{bmatrix} 0 & 1 & 3 & 4 & 8 \end{bmatrix} \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & -\frac{1}{4} \end{bmatrix} \begin{bmatrix} 0 \\ 8 \\ 8 \\ 20 \\ 36 \end{bmatrix} = \begin{bmatrix} 40 \end{bmatrix}.$$

Partition of Observation Equations

Let be given a set of observation equations $A\boldsymbol{x} = \boldsymbol{b} - \boldsymbol{\epsilon}$ partitioned into

$$\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix} - \begin{bmatrix} \boldsymbol{\epsilon}_1 \\ \boldsymbol{\epsilon}_2 \end{bmatrix} \qquad \text{with weights} \qquad \begin{bmatrix} \Sigma_1^{-1} \\ \Sigma_2^{-1} \end{bmatrix}. \tag{2}$$

The matrix A_1 describes usual geodetic observations while A_2 exclusively deals with observations of coordinates. Considering only the first block row of observations we get the usual solution $\hat{\boldsymbol{x}}_1 = \hat{\boldsymbol{x}}_f$:

$$\hat{\boldsymbol{x}}_f = (A_1^{\mathrm{T}} \Sigma_1^{-1} A_1)^{-1} A_1^{\mathrm{T}} \Sigma_1^{-1} \boldsymbol{b}_1 = \Sigma_f A_1^{\mathrm{T}} \Sigma_1^{-1} \boldsymbol{b}_1.$$

The subscript f stands for *free*: The $\hat{\boldsymbol{x}}_f$ solution relates to a network not constrained to the observed coordinates $\boldsymbol{b}_2$. Of course we suppose Σ_f exists!

Next we look for the solution $\boldsymbol{x}_c$ when we include the second block row, namely the coordinate observations $A_2\boldsymbol{x} = \boldsymbol{b}_2 - \boldsymbol{\epsilon}_2$. By adding the normals from both rows we get

$$(A_1^{\mathrm{T}}\Sigma_1^{-1}A_1 + A_2^{\mathrm{T}}\Sigma_2^{-1}A_2)\hat{\boldsymbol{x}} = A_1^{\mathrm{T}}\Sigma_1^{-1}\boldsymbol{b}_1 + A_2^{\mathrm{T}}\Sigma_2^{-1}\boldsymbol{b}_2. \tag{3}$$

We recall that $\Sigma_f^{-1} = A_1^{\mathrm{T}}\Sigma_1^{-1}A_1$ and $A_1^{\mathrm{T}}\Sigma_1^{-1}\boldsymbol{b}_1 = \Sigma_f^{-1}\hat{\boldsymbol{x}}_f$:

$$\hat{\boldsymbol{x}}_c = \left(\Sigma_f^{-1} + A_2^{\mathrm{T}}\Sigma_2^{-1}A_2\right)^{-1}\left(\Sigma_f^{-1}\hat{\boldsymbol{x}}_f + A_2^{\mathrm{T}}\Sigma_2^{-1}\boldsymbol{b}_2\right).$$

The common step in all derivations is our favorite matrix identity

$$\Sigma = (T + A^{\mathrm{T}}\Sigma_e^{-1}A)^{-1} = T^{-1} - T^{-1}A^{\mathrm{T}}(\Sigma_e + AT^{-1}A^{\mathrm{T}})^{-1}AT^{-1}. \tag{4}$$

With (4) we get

$$\hat{\boldsymbol{x}}_c = \left(\Sigma_f - \Sigma_f A_2^{\mathrm{T}}(\Sigma_2 + A_2\Sigma_f A_2^{\mathrm{T}})^{-1}A_2\Sigma_f\right)\left(\Sigma_f^{-1}\hat{\boldsymbol{x}}_f + A_2^{\mathrm{T}}\Sigma_2^{-1}\boldsymbol{b}_2\right).$$

Multiplying out yields

$$\begin{aligned}\hat{\boldsymbol{x}}_c = \hat{\boldsymbol{x}}_f + \Sigma_f A_2^{\mathrm{T}}\Sigma_2^{-1}\boldsymbol{b}_2 - \Sigma_f A_2^{\mathrm{T}}(\Sigma_2 + A_2\Sigma_f A_2^{\mathrm{T}})^{-1}A_2\hat{\boldsymbol{x}}_f \\ - \Sigma_f A_2^{\mathrm{T}}(\Sigma_2 + A_2\Sigma_f A_2^{\mathrm{T}})^{-1}A_2\Sigma_f A_2^{\mathrm{T}}\Sigma_2^{-1}\boldsymbol{b}_2.\end{aligned}$$

Rearranging we get

$$\begin{aligned}\hat{\boldsymbol{x}}_c = \hat{\boldsymbol{x}}_f - \Sigma_f A_2^{\mathrm{T}}(\Sigma_2 + A_2\Sigma_f A_2^{\mathrm{T}})^{-1}A_2\hat{\boldsymbol{x}}_f \\ + \left(\Sigma_f - \Sigma_f A_2^{\mathrm{T}}(\Sigma_2 + A_2\Sigma_f A_2^{\mathrm{T}})^{-1}A_2\Sigma_f\right)A_2^{\mathrm{T}}\Sigma_2^{-1}\boldsymbol{b}_2.\end{aligned}$$

Using (4) once again—in the opposite direction—we get

$$\hat{\boldsymbol{x}}_c = \hat{\boldsymbol{x}}_f - \Sigma_f A_2^{\mathrm{T}}(\Sigma_2 + A_2\Sigma_f A_2^{\mathrm{T}})^{-1}A_2\hat{\boldsymbol{x}}_f + \left(\Sigma_f^{-1} + A_2^{\mathrm{T}}\Sigma_2^{-1}A_2\right)^{-1}A_2^{\mathrm{T}}\Sigma_2^{-1}\boldsymbol{b}_2.$$

Finally the matrix identity (26) takes us the last step:

$$\hat{\boldsymbol{x}}_c = \hat{\boldsymbol{x}}_f - \Sigma_f A_2^{\mathrm{T}}(\Sigma_2 + A_2\Sigma_f A_2^{\mathrm{T}})^{-1}A_2\hat{\boldsymbol{x}}_f + \Sigma_f A_2^{\mathrm{T}}(\Sigma_2 + A_2\Sigma_f A_2^{\mathrm{T}})^{-1}\boldsymbol{b}_2$$

or rearranged

$$\hat{\boldsymbol{x}}_c = \hat{\boldsymbol{x}}_f + \Sigma_f A_2^{\mathrm{T}}(\Sigma_2 + A_2\Sigma_f A_2^{\mathrm{T}})^{-1}(\boldsymbol{b}_2 - A_2\hat{\boldsymbol{x}}_f). \tag{5}$$

From (3) we read the covariance matrix of the *constrained* estimate $\hat{\boldsymbol{x}}_c$

$$\Sigma_c = (A_1^{\mathrm{T}}\Sigma_1^{-1}A_1 + A_2^{\mathrm{T}}\Sigma_2^{-1}A_2)^{-1}. \tag{6}$$

We recall that $\Sigma_f = (A_1^{\mathrm{T}} \Sigma_1^{-1} A_1)^{-1}$, so (6) can be written

$$\Sigma_c = (\Sigma_f^{-1} + A_2^{\mathrm{T}} \Sigma_2^{-1} A_2)^{-1}. \tag{7}$$

Using (4) we get

$$\Sigma_c = \Sigma_f - \Sigma_f A_2^{\mathrm{T}} (\Sigma_2 + A_2 \Sigma_2 A_2^{\mathrm{T}})^{-1} A_2 \Sigma_f. \tag{8}$$

A nonnegative definite matrix is subtracted from Σ_f, so

$$\Sigma_c \leq \Sigma_f. \tag{9}$$

This says that ***covariances in the constrained network are smaller than or equal to those in the free network***.

Putting $\Sigma_2 = 0$ in (2)—the observation error equals zero—is equivalent to introduce the constraint $A_2 \boldsymbol{x} = \boldsymbol{b}_2$:

$$A_1 \boldsymbol{x} = \boldsymbol{b}_1 - \boldsymbol{\epsilon}_1 \tag{10}$$

$$A_2 \boldsymbol{x} = \boldsymbol{b}_2. \tag{11}$$

This has a large impact on formulation of filters. Krarup calls this a *hard postulation* of the least-squares problem, opposed to a *soft postulation* of the constraints with $\Sigma_2 > 0$.

So setting the observation error equal to zero in a Kalman filter is the same as using a constraint in a standard least-squares formulation.

Dependency on the Weights

$$\boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix}, \ \boldsymbol{\epsilon} = \begin{bmatrix} \boldsymbol{\epsilon}_1 \\ \boldsymbol{\epsilon}_2 \end{bmatrix}, \ A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, \ \Sigma^{-1} = \begin{bmatrix} \Sigma_1^{-1} & 0 \\ 0 & \Sigma_2^{-1} \end{bmatrix}.$$

The normal equation for the original, total problem is

$$\big(A_1^{\mathrm{T}}\Sigma_1^{-1}A_1 + A_2^{\mathrm{T}}\Sigma_2^{-1}A_2\big)\hat{\boldsymbol{x}} = A_1^{\mathrm{T}}\Sigma_1^{-1}\boldsymbol{b}_1 + A_2^{\mathrm{T}}\Sigma_2^{-1}\boldsymbol{b}_2. \tag{12}$$

The perturbed problem is

$$\begin{aligned}\big(A_1^{\mathrm{T}}\Sigma_1^{-1}A_1 + A_2^{\mathrm{T}}(\Sigma_2^{-1} + (\Delta\Sigma_2^{-1}))A_2\big)(\hat{\boldsymbol{x}} + \Delta\boldsymbol{x}) = \\ \big(A_1^{\mathrm{T}}\Sigma_1^{-1}\boldsymbol{b}_1 + A_2^{\mathrm{T}}(\Sigma_2^{-1} + (\Delta\Sigma_2^{-1}))\boldsymbol{b}_2\big).\end{aligned} \tag{13}$$

Now subtract (12) from (13) to find an equation for $\Delta\boldsymbol{x}$:

$$\big(A_1^{\mathrm{T}}\Sigma_1^{-1}A_1 + A_2^{\mathrm{T}}(\Sigma_2^{-1} + (\Delta\Sigma_2^{-1}))A_2\big)\Delta\boldsymbol{x} + A_2^{\mathrm{T}}(\Delta\Sigma_2^{-1})A_2\hat{\boldsymbol{x}} = A_2^{\mathrm{T}}(\Delta\Sigma_2^{-1})\boldsymbol{b}_2.$$

We set $N = A_1^{\mathrm{T}} \Sigma_1^{-1} A_1 + A_2^{\mathrm{T}} \Sigma_2^{-1} A_2$ and $\hat{\boldsymbol{\epsilon}}_2 = \boldsymbol{b}_2 - A_2 \hat{\boldsymbol{x}}$. Then the change in $\hat{\boldsymbol{x}}$ is

$$\Delta \boldsymbol{x} = \big(N + A_2^{\mathrm{T}} (\Delta \Sigma_2^{-1}) A_2\big)^{-1} A_2^{\mathrm{T}} (\Delta \Sigma_2^{-1}) \hat{\boldsymbol{\epsilon}}_2. \tag{14}$$

Now (4) yields the change in the inverse:

$$\big(N + A_2^{\mathrm{T}} (\Delta \Sigma_2^{-1}) A_2\big)^{-1} = N^{-1} - N^{-1} A_2^{\mathrm{T}} \big((\Delta \Sigma_2^{-1})^{-1} + A_2 N^{-1} A_2^{\mathrm{T}}\big)^{-1} A_2 N^{-1}.$$

For m observations, this matrix multiplies $A_2^{\mathrm{T}} (\Delta \Sigma_2^{-1}) \hat{\boldsymbol{\epsilon}}_2$ to give $\Delta \boldsymbol{x}$. If we specialize to one single observation, $\boldsymbol{a}_2^{\mathrm{T}}$ is an m by 1 vector and $\Delta \Sigma_2^{-1}$ is a 1 by 1 scalar. We name the product $\boldsymbol{a}_2 N^{-1} \boldsymbol{a}_2^{\mathrm{T}} = s$ and get

$$\Delta \boldsymbol{x} = N^{-1} \boldsymbol{a}_2^{\mathrm{T}} \left(1 - \frac{\Delta \Sigma_2^{-1}}{1 + s\,(\Delta \Sigma_2^{-1})} s \right) (\Delta \Sigma_2^{-1})\, \hat{\epsilon}_2$$

or

$$\Delta \boldsymbol{x} = \frac{\hat{\epsilon}_2\,(\Delta \Sigma_2^{-1})}{1 + s\,(\Delta \Sigma_2^{-1})} N^{-1} \boldsymbol{a}_2^{\mathrm{T}}. \tag{15}$$

A First Order Derivation

If we consider a *small change* $\Delta\Sigma^{-1}$ in the weighting matrix, we can compute the corresponding small movement $\Delta\hat{\boldsymbol{x}}$ in the estimate. The problem is linearized and higher order terms are ignored:

$$A^{\mathrm{T}}\big(\Sigma^{-1} + (\Delta\Sigma^{-1})\big)A(\hat{\boldsymbol{x}} + \Delta\hat{\boldsymbol{x}}) = A^{\mathrm{T}}\big(\Sigma^{-1} + (\Delta\Sigma^{-1})\big)\boldsymbol{b}.$$

Subtracting $A^{\mathrm{T}}\Sigma^{-1}A\hat{\boldsymbol{x}} = A^{\mathrm{T}}\Sigma^{-1}\boldsymbol{b}$ leaves an equation for the first-order correction $\Delta\hat{\boldsymbol{x}}$:

$$A^{\mathrm{T}}\Sigma^{-1}A(\Delta\hat{\boldsymbol{x}}) = A^{\mathrm{T}}(\Delta\Sigma^{-1})(\boldsymbol{b} - A\hat{\boldsymbol{x}}).$$

This correction to $\hat{\boldsymbol{x}}$ is small when $\Delta\Sigma^{-1}$ is small, and when the residual $\boldsymbol{b} - A\hat{\boldsymbol{x}}$ is small.

If we work with the covariance matrix Σ instead of Σ^{-1}, linearization gives $\Delta\Sigma^{-1} = -\Sigma^{-1}(\Delta\Sigma)\Sigma^{-1}$. This is the matrix analog of $-\Delta x/x^2$, the first-order change in $1/x$.

Linearization also gives the change in $Q = (A^{\mathrm{T}}\Sigma^{-1}A)^{-1}$, which contains the variances and covariances of the estimate $\hat{\boldsymbol{x}}$:

$$\Delta Q = QA^{\mathrm{T}}(\Delta\Sigma^{-1})AQ.$$

Useful Matrix Identities

Whenever the following inverse matrices exist we have

$$(AB)^{-1} = B^{-1}A^{-1} \tag{16}$$

$$(I + AB)^{-1}A = A(I + BA)^{-1} \tag{17}$$

$$A \text{ and } B \text{ may not be square} \tag{18}$$

for vectors we especially have

$$(I + \boldsymbol{a}\boldsymbol{b}^{\mathrm{T}})^{-1}\boldsymbol{a} = \boldsymbol{a}(I + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{a})^{-1} \tag{19}$$

$$(A^{-1} + B^{-1})^{-1} = A(A + B)^{-1}B = B(A + B)^{-1}A \tag{20}$$

$$(A + BDC)^{-1} = A^{-1} - A^{-1}B(D^{-1} + CA^{-1}B)^{-1}CA^{-1} \tag{21}$$

$$DC(A + BDC)^{-1} = DCA^{-1}(I + BDCA^{-1}) \tag{22}$$

$$= DC(I + A^{-1}BDC)^{-1}A^{-1} \tag{23}$$

$$= D(I + CA^{-1}BD)^{-1}CA^{-1} \tag{24}$$

$$= (I + DCA^{-1}B)^{-1}DCA^{-1} \tag{25}$$

$$= (D^{-1} + CA^{-1}B)^{-1}CA^{-1}. \tag{26}$$

MATLAB Code Demonstrating the Theory

```
A = [1 2 6; 2 3 7; 3 4 8; 4 5 10];
b = [0;7;8;20];
x = A\b
A1 = A(:,1:2);
A2 = A(:,3);
%Projector
P = eye(4)-A1*inv(A1'*A1)*A1';
PA = P*A;
Pb = P*b;
x3 = Pb(4)/PA(4,3)
%Block elimination
N = A'*A;
b0 = A'*b;
b1 = b0(1:2);
b2 = b0(3);
A0 = N(1:2,1:2);
B = N(1:2,3);
D = N(3,3);
x3 = inv(D-B'*inv(A0)*B)*(b2-B'*inv(A0)*b1)
%General formula
M = A2'*A1*inv(A1'*A1)*A1';
R = A2'-M;
x3 = inv(R*A2)*R*b
```

Reference

Strang, Gilbert & Kai Borre (1997) *Linear Algebra, Geodesy, and GPS*. Wellesley-Cambridge Press. Order at **www.wellesleycambridge.com**