Real-Time Applications

Especially the surveying community has tried to push some boundaries of differential GPS processing as described up till now. Surveyors want coordinates in real time and with good accuracy; this means standard deviations of all three coordinates at centimeter level. Actually this is possible.

We need differential GPS, that is a static master receiver and one or more roving receivers. The observational data have to be collected at the same location for processing. Most often this happens at the rover receiver; this is convenient if we want to use coordinates for immediate set out of new points. So in principle we transfer the data processing from a postprocessing situation on a personal computer to a processor at the rover. For this to take place we need to transfer the observations from the master receiver to the rover receiver. This is most often done via a telemetry link with a speed of 4 800 or 9 600 bits per second. The format for the data transfer is vendor dependent. However, in the later years a receiver independent format for this sort of data has been developed. It is known as the RTCM format, see RTCM (1998); the current version is 2.2.

In Table 15.1 we list the main data types defined. For our purpose types 18, 19, 20, and 21 are relevant. Types 10, 11 are used for less demanding purposes. The use of a telemetry link demands line of sight between master and rover receivers. If this is not the case you may place signal repeaters at high points in the terrain. A more severe problem with data transfer in real time is the delay of the transferred data. The electronic delays in the system and a more important part is the delay caused by the distance between the two receivers. The only way to deal with this problem is to predict the observations to what they would be at the rover site were they not delayed. Hence our method only can become a *near* real-time application. The computed position at the rover is only valid for the epoch time advanced with processing time.

Real-time Positioning Using Differential Carrier Phase

Most published models for computing real-time differential corrections for the rover are based on simultaneity. The corrections calculated at the master i are directly transmitted to the rover j. But this procedure inherits especially a latency problem.

We repeat the basic equation for a phase observation using satellite k:

$$\Phi_i^k(t) = \rho_i^k - I_i^k + T_i^k + c\bigl(dt^k(t-\tau_i^k) - dt_i(t)\bigr) + \lambda\bigl(\varphi_i(t_0) - \varphi^k(t_0)\bigr) + \lambda N_i^k + \epsilon_i^k.$$

Double frequency receivers will eliminate the ionospheric delay I_i^k. The tropospheric delay T_i^k can be modeled and largely removed. Anyway the distance between master and rover can never extend the distance over which the correction signal can be transmitted. This is typically less than 25 km. The tropospheric correction over such distances nearly cancels out in double differences. We also assume the ambiguity N_i^k to be solved on-the-fly (typical duration 15–30 s).

All information about a receiver position and dynamics is contained in the range ρ_i^k. The remaining terms on the right generally are unknown and need to be accounted for in

Table 15.1 Message types

No.	Status	Title
1	fixed	differential GPS corrections
2	fixed	delta differential GPS corrections
3	fixed	GPS reference station parameters
4	tentative	reference station datum
5	fixed	GPS constellation health
6	fixed	GPS null frame
7	fixed	DGPS radiobeacon almanac
8	tentative	pseudolite almanac
9	fixed	GPS partial correction set
10	reserved	P-code differential corrections
11	reserved	C/A-code L_1, L_2 delta corrections
12	reserved	pseudolite station parameters
13	tentative	ground transmitter parameters
14	tentative	GPS time of week
15	tentative	ionospheric delay message
16	fixed	GPS special message
17	tentative	GPS ephemerides
18	fixed	RTK uncorrected carrier phases
19	fixed	RTK uncorrected pseudoranges
20	tentative	RTK carrier phase corrections
21	tentative	RTK high accuracy pseudorange corrections
22	tentative	extended reference station parameters
23–30	—	undefined
31	tentative	differential GLONASS corrections
32	tentative	differential GLONASS reference station parameters
33	tentative	GLONASS constellation health
34	tentative	GLONASS partial differential correction set ($N > 1$), GLONASS null frame ($N \leq 1$)
35	tentative	GLONASS radiobeacon almanac
36	tentative	GLONASS special message
37	tentative	GNSS system time offset
38–58	—	undefined
59	fixed	proprietary message
60–63	reserved	multipurpose usage

the processing. However at the master we can determine a phase correction:

$$\Phi^k_{c,i}(t) = \Phi^k_i - \big(\rho^k_i + c\,dt^{\text{Broadcast},k}(t - \tau^k_i) - c\,dt_i(t)\big). \tag{15.65}$$

The approximate master clock offset $c\,dt_i(t)$ is computed using an appropriate algorithm. The phase correction $\Phi^k_{c,i}(t)$ is thus equivalent to

$$\Phi^k_{c,i}(t) = c\,dt^{\mathrm{SA},k}(t) - I^k_i + T^k_i + \lambda\bigl(\varphi_i(t_0) - \varphi^k(t_0)\bigr) + \lambda N^k_i + \epsilon^k_i. \tag{15.66}$$

The term $c\,dt^{\mathrm{SA},k}(t) = c\,dt^k(t-\tau^k_i) - c\,dt^{\mathrm{Broadcast},k}(t-\tau^k_i)$ represents the unknown satellite clock dithering due to SA (selective availability).

The phase corrections given in (15.65) and (15.66) refer to the past time $t = t_0$. But these corrections have to be extrapolated to the current user time t_e. Any second order extrapolation involves errors that depend on the correction rates and their accelerations. The changes in the observed carrier phase are mainly due to the satellite clock dithering $c\,dt^{\mathrm{SA},k}(t)$. The orbit and atmosphere errors vary slowly by comparison.

Experiments show that the correction accelerations can be as large as 0.01 m/s^2. Neglecting these accelerations would cause an error of several centimeters in the extrapolation. So the second order extrapolation model must be as follows:

$$\Phi^k_{c,i}(t_e) = \Phi^k_{c,i}(t_0) + \dot{\Phi}^k_{c,i}(t_0)(t_e - t_0) + \tfrac{1}{2}\ddot{\Phi}^k_{c,i}(t_0)(t_e - t_0)^2. \tag{15.67}$$

The phase rate $\dot{\Phi}^k_{c,i}(t_0)$ and the acceleration $\ddot{\Phi}^k_{c,i}(t_0)$ are estimated from the past observations at the master i. At the rover j one applies the extrapolated corrections (15.67):

$$\widetilde{\Phi}^k_j(t_e) = \Phi^k_j(t_e) - \Phi^k_{c,i}(t_e). \tag{15.68}$$

The actual observable used in the rover differential phase positioning filter is the difference of the corrected phase observations with respect to a reference satellite k. The observation model for the rover is thus derived by combining equations (15.65) and (15.68):

$$\widetilde{\Phi}^{kl}_j(t_e) = \rho^{kl}_j(t_e) + N^{kl}_{ij}(t_e). \tag{15.69}$$

If $t_0 = t_e$ the formulation in (15.69) is equivalent to the double difference kinematic model:

$$\Phi^{kl}_{ij}(t_0) = \rho^{kl}_{ij}(t_0) + N^{kl}_{ij}(t_0).$$

Experiments show that *the phase prediction error is on average below 5 cm at a correction update interval of 5 seconds*. At this rate the positioning accuracy should be maintained at the several-centimeter level. This opens the possibility of using slower data links than those required for real time kinematic positioning to maintain the same position output rate. Applications of differential phase positioning include construction machine guidance and high resolution hydrographic surveying, where continuous output with minimum latency is a must.

Apart from the problems already delt with there are a few additional issues for the real-time application. The epoch interval has to be 1 second or shorter. So an update rate of 1–2 Hz is essential else the user experiences that the system has a slow performance and he gets annoyed. This update rate of course puts high demands on the processor.

One way of processing the data is through a filter for double differenced data. To achive centimeter accuracy the state vector must contain the coordinate differences (x, y, z)

between master and rover, the L_1 ambiguities N_1, the L_2 ambiguities N_2, and if the baseline is longer than a few kilometers also ionospheric delays I for each satellite. Some people prefer a filter for single differenced data. In that case we have to augment the state vector with the receiver clock bias.

For DD data from s satellites we have for one epoch 3 coordinate differences, $s-1$ N_1-values, $s-1$ N_2-values that is $2s+1$ unknowns but only $2s$ observations. So initially we need at least two epochs for a meaningful estimation. If we do not move from epoch 1 to epoch 2, the situation after epoch 2 is the following: $4s$ observations and $2s+1$ unknowns. In order to get initial values for the filter we collect observations from a few epochs and add them to a set of normals. The solution of these normals act as the initial guess for the state vector. The next issue is to decide when to try to switch the *real valued* ambiguities into *integers*. Recently Teunissen indicated how to compute the probability for finding the correct sets of integer ambiguities. As soon as this is larger than 0.99 we make a lambda call.

If everything works perfectly we now have a filter that runs smoothly and the changes in coordinates of the state vector are at mm level. However we have to specify covariances Σ_e, and Σ_ϵ, and F.

Explicit description of DD-filter

Explicit description for SD-filter

Ambiguities and Real-Time Filters

Let the state vector be split into a part with coordinate differences x_c and a part with ambiguities x_N:

$$\boldsymbol{x} = \begin{bmatrix} \hat{x}_c \\ \hat{x}_N \end{bmatrix}.$$

At some time we make the call lambda. The output is the integers $\bar{x}_N$ and the two values disall(1) and disall(2). Later on we return to the question when to make that call.

Now we may correct the state vector to include the integer values of the ambiguities (still we are not sure if they are the correct ones. Later on we also return to this problem!):

$$\boldsymbol{x} = \begin{bmatrix} \hat{x}_c \\ \bar{x}_N \end{bmatrix}. \tag{15.70}$$

However this change has an impact on both $\hat{x}_c$ and the corresponding covariance matrix. Let the total covariance matrix be

$$\Sigma = \begin{bmatrix} \Sigma_{\hat{x}_c} & \Sigma_{\hat{x}_c \hat{x}_N} \\ \Sigma_{\hat{x}_N \hat{x}_c} & \Sigma_{\hat{x}_N} \end{bmatrix}$$

then we have to modify the x_c part of the state vector as follows

$$\bar{x}_c = \hat{x}_c - \Sigma_{\hat{x}_c \hat{x}_N} \Sigma_{\hat{x}_N}^{-1} \Sigma_{\hat{x}_N \hat{x}_c} (\hat{x}_N - \bar{x}_N) \tag{15.71}$$

and the covariance matrix is modified as follows

$$\Sigma_{\bar{x}_c} = \Sigma_{\hat{x}_c} - \Sigma_{\hat{x}_c \hat{x}_N} \Sigma_{\hat{x}_N}^{-1} \Sigma_{\hat{x}_N \hat{x}_c}. \tag{15.72}$$

Equations (15.70) to (15.72) are theoretically correct, but in practice one often uses an alternative technique based on the following trick:

$$\bar{x}_N = \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} \hat{x}_c \\ \hat{x}_N \end{bmatrix}.$$

We use $\Sigma_{\bar{x}_N} = \sigma_N^2 I$ with e.g. $\sigma_N = 10^{-5}$ cycle.

When to fix the ambiguities? In geodetic processing of GPS data there has been a long tradition to consider a ratio called "best to next-best variance". If the sum of squares for float

$$s = \boldsymbol{r}^{\mathrm{T}} \Sigma^{-1} \boldsymbol{r}$$

the ***ratio*** is defined as

$$\text{ratio} = \frac{s + \mathsf{disall}(1)}{s + \mathsf{disall}(2)}. \tag{15.73}$$

The disall-values are provided by calling lambda. If square root of ratio is 1.2–1.4 or larger, it is worthwhile to try to fix the floating ambiguities.

Today there are theoretically well-founded methods described in Teunissen (1998). If $\sqrt[n]{\det(\Sigma_{\hat{N}})} < 0.2$ cycle, then it is time to make the first lambda call for fixing the ambiguities. The criterion secures with a probability of 0.98 to have fixed the correct ambiguities. (Since 1992 the Ashtech software GPPS has used the limit 0.95).

REFERENCES

RTCM (1998). RTCM Recommended standards for differential GNSS (Global Navigation Satellite Systems) service. RTCM PAPER 11-98/SC104-STD, Radio Technical Commission For Maritime Services, 1 800 Diagonal Road, Suite 600, Alexandria, Virginia 22314-2840, U. S. A.

Teunissen, P. J. G. (1998). Success probability of integer GPS ambiguity rounding and bootstrapping. *Journal of Geodesy*, 72:606–612.