The Aalborg GPS Software Defined Radio Receiver

Kai Borre

Danish GPS Center, Aalborg University
Niels Jernes Vej 14, DK-9220 Aalborg Ø, Denmark
Phone: +4596358362, e-mail: borre@gps.aau.dk

Abstract. A receiver for the Global Positioning System (GPS) signals provides information on its position and time. The position is given in an Earth-Centered and Earth-Fixed coordinate system. This means that a static receiver keeps its coordinates over time, apart from the influence of measurement errors. The system time (GPST) counts in weeks and seconds of week starting on January 6, 1980. Each week has its own number. Time within a week is counted in seconds from the beginning at midnight between Saturday and Sunday (day 1 of the week). GPST is maintained within the system itself. Universal Time Coordinated (UTC) goes at a different rate which is connected to the actual speed of the rotation of the Earth. At present 14 seconds have to be added to UTC to get GPST.

The GPS has 6 orbital planes with at least 4 satellites. At the moment GPS consists of 29 active satellites. They complete about 2 orbits/day.

1 The Transmitted GPS Signals

Satellite positioning systems exploit ***Spread Spectrum*** (SS) techniques. SS came alive in 1980s and is popular for applications involving radio links in hostile environments. SS is an RF communications system in which the baseband signal bandwidth is intentionally spread over a larger bandwidth by injecting a higher-frequency signal. As a direct consequence, energy used in transmitting the signal is spread over a wider bandwidth and appears as noise. The ratio (in dB) between the spread baseband and the original signal is called *processing gain*. Typical SS processing gains run from 10 dB to 60 dB, see [1].

To apply an SS technique, simply inject the corresponding SS code somewhere in the transmitting chain before the antenna. That injection is called the spreading operation. The effect is to diffuse the information in a larger bandwidth. Conversely, you can remove the SS code by a despreading operation, at a point in the receive chain before data retrieval. The effect of a despreading operation is to reconstitute the information in its original bandwidth. Obviously, the same code must be known in advance at both ends of the transmission channel.

In GPS, SS modulation is applied on top of a BPSK modulation, see below.

Intentional or un-intentional interference and jamming signals are rejected because they do not contain the SS code. This characteristic is the real beauty of SS. Only the desired signal, which has the code, will be seen at the receiver when the despreading operation is exercised.

In GPS the codes are digital sequences that must be as long and as random as possible to appear as "noise-like" as possible. But in any case, they must remain reproducible. Otherwise, the receiver will be unable to extract the message that has been sent. Thus, the sequence is "nearly random". Such a code is called a ***pseudo-random number*** (PRN) or sequence. The PRN sequences applied in GPS are ***Gold sequences***. These sequences are generated by feedback shift registers, and they are inserted at the data level. This is the direct sequence form of spread spectrum (DSSS). The PRN is applied directly to data entering the carrier modulator.

All GPS satellites use the same carrier frequencies: On L_1 1575.42 MHz and L_2 1227.60 MHz. In a modernized GPS there will be a new civilian frequency L_5 (then the military might remove L_2 from civilian use).

Each satellite has two unique spreading sequences or codes. The first one is the coarse acquisition code (C/A) and the other one is the encrypted precision code (P(Y)). The C/A code is a sequence of 1 023 chips. (A chip corresponds to a bit. It is simply called a chip to emphasize that it does not hold any information.) The code is repeated each ms giving a chipping rate of 1.023 MHz. The P code is a longer code ($\approx 2.35 \cdot 10^4$ chips) with a chipping rate of 10.23 MHz. It repeats itself each week starting at the beginning of the GPS week. The C/A code is only modulated onto the L_1 carrier while the P(Y) code is modulated onto both the L_1 and the L_2 carrier.

The purpose of PRN codes is twofold: They spread the signals and they provide for measuring the travel time between satellite and receiver. The system keeps all C/A code starts aligned in all active satellites.

In the rest of this presentation we focus on the L_1 signal. Each satellite transmits a continuous signal with at least three components:

- a carrier wave with frequency $f_1 = 1575.42 = 154 \times 10.23$ MHz
- an individual PRN code which is a sequence of −1 and +1 each of length 1 millisecond
- a data bit sequence which carries information from which the satellite's position can be computed. The length of one navigation bit is 20 milliseconds.

The PRN code and the data bits are combined through modulo-2 adders. The result is modulated onto the carrier signal using the ***binary phase shift keying*** (BPSK) method: The carrier is instantaneously phase shifted by 180° at the time of a chip change. When a navigation data bit transition occurs, the phase of the resulting signal is also phase shifted 180°.

So the ***signal transmitted*** from satellite k is

$$
\begin{aligned}
s^k(t) = {} & \sqrt{2P_C}\left(C^k(t) \oplus D^k(t)\right)\cos(2\pi f_{L1} t) \\
& + \sqrt{2P_{PL1}}\left(P^k(t) \oplus D^k(t)\right)\sin(2\pi f_{L1} t) \\
& + \sqrt{2P_{PL2}}\left(P^k(t) \oplus D^k(t)\right)\sin(2\pi f_{L2} t). \qquad (1)
\end{aligned}
$$

Here P_C, P_{PL1}, and P_{PL2} are the powers of signals with C/A or P code. C^k and P^k are the C/A and P(Y) code sequences assigned to satellite number k. D^k is the navigation data sequence, and f_{L1} and f_{L2} are the carrier frequencies of L1 and L2. The ⊕ symbol denotes the "exclusive or" operation.

2 The Received GPS Signals

Let the total received power be P, and let the transmission delay (traveling time) be τ. The carrier frequency offset is Δf (Doppler), and the received phase is θ. Then the *received* L_1 *signal* can be written as

$$S^k(t) = \text{const.} \times \sqrt{2P}\, D^k(t-\tau)\cos(2\pi(f-\Delta f)(t-\tau)+\theta). \tag{2}$$

The data coefficient D^k is a product of code sequences and navigation data for satellite k.

From the observation $s(t)$ we want to estimate τ, Δf, and θ. This is done in a two step procedure

1. find global approximate values of τ and Δf, called ***signal acquisition***
2. local search for τ, Δf and possibly the carrier phase θ:
   - If θ is estimated the search is called *coherent signal tracking*.
   - If θ is ignored, the search is called ***non-coherent signal tracking***.

The purpose of code tracking is to estimate the travel time τ. This is done by means of a *delay lock loop* (DLL). For a coherent DLL we have $\theta = 0$.

To demodulate the navigation data, a carrier wave replica must be generated.

To track a carrier wave signal, a *phase lock loop* (PLL) often is used.

3 Receiver Channels and Acquisition

The signal processing for satellite navigation systems is based on a channelized structure. Next, we provide an overview of the concept of a ***receiver channel*** and the processing that occurs.

Figure 1 gives an overview of a channel. Before allocatting a channel to a satellite, the receiver must know which satellites that are currently visible.

The received signal $s(t)$ is a combination of signals from all n visible satellites

$$s(t) = s^1(t) + s^2(t) + \cdots + s^n(t). \tag{3}$$

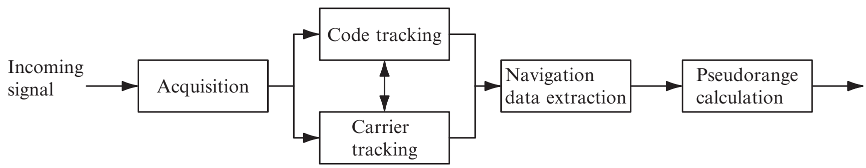

Fig. 1. One receiver channel. The acquisition gives rough estimates of signal parameters. These parameters are refined by the two tracking blocks. After tracking, the navigation data can be extracted and pseudoranges can be computed.

Before dealing specifically with satellite k we allocate a *channel* to acquiring it. This happens by using the following steps:

- the incoming signal s is multiplied with the locally generated C/A code corresponding to the satellite k

  The cross-correlation between C/A codes for different satellites implies that signals from other satellites are nearly removed by this procedure. To avoid removing the desired signal component, the locally generated C/A code must be properly aligned in time, that is have the correct code phase.
- After multiplication with the locally generated code, the signal must be mixed with a locally generated carrier wave; this removes the carrier wave of the received signal.

  In order to do this successfully the frequency of the locally generated signal must be close to the signal carrier frequency.

Next all signal components are squared and summed providing a numerical value. The acquisition procedure is a ***search procedure.***

It is sufficient to search Δf in steps of 500 Hz in the interval ±10 kHz. There are 1 023 discrete values of the code phase. A search for the maximum value over this 41 × 1 023 grid is performed either as

1. Parallel ***frequency space search*** acquisition (search 1 023 different code phases), see Fig. 2:
   a) The incoming signal is multiplied with a locally generated C/A code for satellite k
   b) the result is FFT from time domain to frequency domain
   c) absolute values of all components are computed
   d) if satellite k is present we can identify a maximum value at (code phase, frequency) = $(\tau, f - \Delta f)$. An unsuccessful search is shown in Fig. 3(a) and a successful one in Fig. 3(b),

or as

2. Parallel ***code phase search*** acquisition (search 41 carrier frequencies) (Fig. 4):
   a) Multiply incoming signal with cosine or sine, giving I and Q
   b) combine I and Q to complex input to FFT
   c) generate local C/A code for satellite k, FFT, complex conjugate and multiply with output from (b)
   d) IFFT and compute absolute values of all components
   e) maximum value at (code phase τ, frequency $f - \Delta f$) if satellite k is present. Else no distinct maximum value.

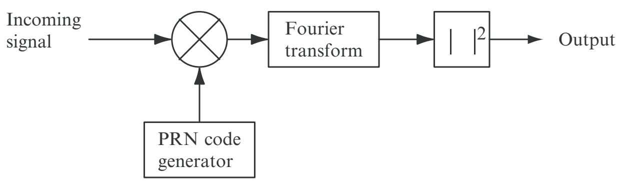

Fig. 2. Block diagram of the parallel *frequency space search* algorithm.

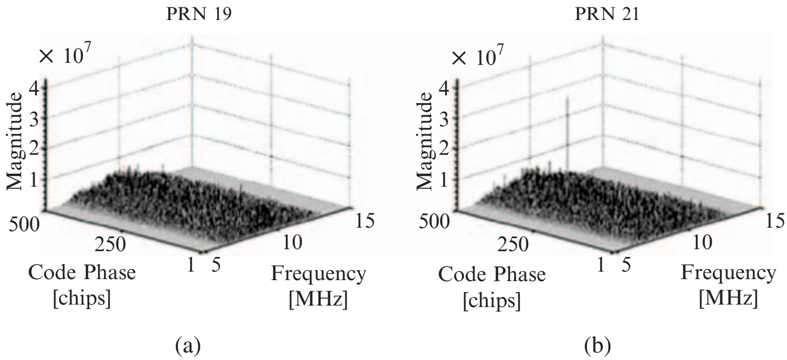

Fig. 3. Output from parallel frequency space search acquisition. The figure only includes the first 500 chip shifts and the frequency band from 5–15 MHz. (a) PRN 19 is not visible so no significant peaks are present in the spectrum. (b) PRN 21 is visible so a significant peak is present in the spectrum. The peak is situated at code phase 359 chips and frequency 9.548 MHz.

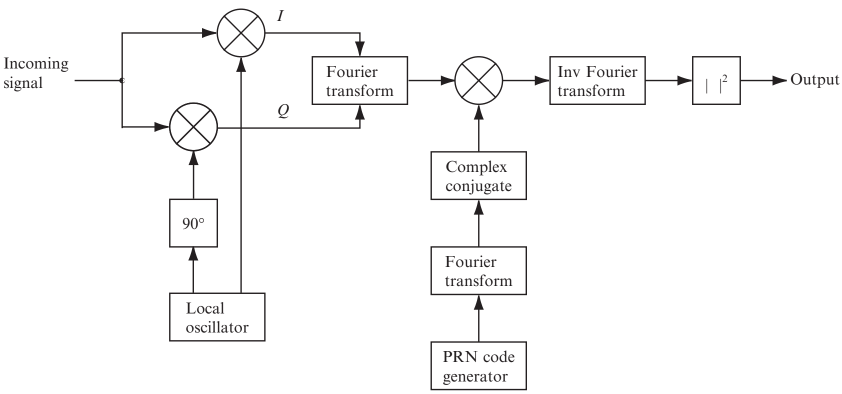

Fig. 4. Block diagram of the parallel *code phase search* algorithm.

In our implementation the code phase search is ten times faster than frequency space search.

4 Carrier and Code Tracking

The carrier tracking (phase lock loop PLL) involves the following issues:

– Improve the estimate of the carrier frequency obtained by acquisition
– generate local carrier signal
– measure phase error between incoming carrier and local carrier signal
– adjust frequency until phase and frequency become stable.

To demodulate the navigation data successfully a carrier wave replica has to be generated. To track a carrier wave signal ***Phase Lock Loops*** (PLL) or Frequency Lock Loops (FLL) are often used.

Figure 5 shows a basic block diagram for a phase lock loop. The two first multiplications wipe off the carrier and the PRN code of the input signal. To wipe off the PRN code, the I_p output from the early-late code tracking loop described above is used. ***The loop discriminator block is used to find the phase error*** on the local carrier wave replica. The output of the ***discriminator***, which is the phase error ϕ (or a function of the phase error), is then filtered and used as a feedback to the Numerically Controlled Oscillator (NCO) which adjusts the frequency of the local carrier wave. In this way the local carrier wave could be an almost precise replica of the input signal carrier wave.

The problem with using an ordinary PLL is that it is sensitive to 180° phase shifts. The PLL used in a GPS receiver has to be insensitive to 180° phase shifts due to navigation bit transitions,

The Costas loop is insensitive for 180° phase shifts. The Costas loop in Fig. 6 contains two multiplications. The first multiplication is the product between the input signal and the local carrier wave and the second multiplication is between a 90° phase shifted carrier wave and the input signal. The ***goal of the Costas loop is to try to keep all energy in the I (in-phase) arm.*** To keep the energy in the ***I*** arm some kind of feedback to the oscillator is needed. If the code replica in Fig. 6 is perfectly aligned, the multiplication in the ***I*** arm yields the following sum

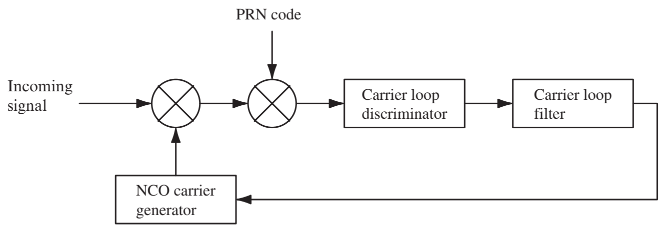

Fig. 5. Basic GPS receiver tracking loop block diagram.

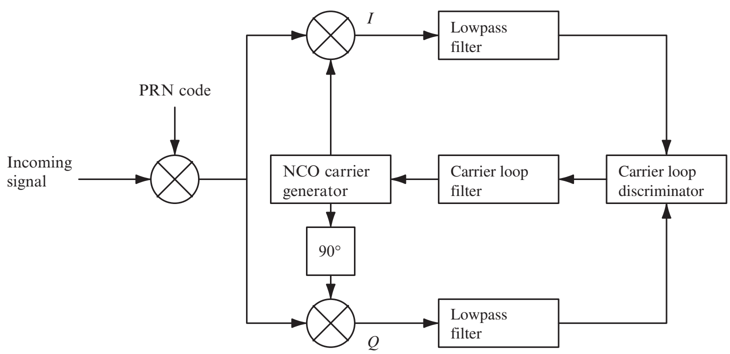

Fig. 6. Costas loop used to track the carrier wave.

$$D^k(n)\cos(\omega_{\mathrm{IF}}n)\cos(\omega_{\mathrm{IF}}n+\phi)=\frac{1}{2}D^k(n)\cos(\phi)+\frac{1}{2}D^k(n)\cos(2\omega_{\mathrm{IF}}n+\phi) \quad (4)$$

where ϕ is the phase difference between the phase of the input signal and the phase of the local replica of the carrier phase. The multiplication in the quadrature arm gives the following

$$D^k(n)\cos(\omega_{\mathrm{IF}}n)\sin(\omega_{\mathrm{IF}}n+\phi)=\frac{1}{2}D^k(n)\sin(\phi)+\frac{1}{2}D^k(n)\sin(2\omega_{\mathrm{IF}}n+\phi). \quad (5)$$

If the two signals are lowpass filtered after the multiplication, the two terms with $(2\omega_{\mathrm{IF}}n + \phi)$ are eliminated and the following two signals remain

$$I^k=\frac{1}{2}D^k(n)\cos(\phi) \quad (6)$$

$$Q^k=\frac{1}{2}D^k(n)\sin(\phi). \quad (7)$$

To define a quantity to feedback to the carrier phase oscillator, the phase error ϕ of the local carrier phase replica is a good candidate which can be found as

$$\frac{Q^k}{I^k}=\frac{\frac{1}{2}D^k(n)\sin(\phi)}{\frac{1}{2}D^k(n)\cos(\phi)}=\tan(\phi) \quad (8)$$

$$\phi=\tan^{-1}\left(\frac{Q^k}{I^k}\right). \quad (9)$$

From equation (9) it can be seen that the phase error is small when the correlation in the quadrature-phase arm is close to zero and the correlation value in the in-phase arm is maximum.

The goal of a code tracking loop is to keep track of the phase of a specific code in the signal. The output of such a code tracking loop is a perfectly aligned replica of the code. The code tracking loop in the GPS receiver is a delay lock loop (DLL) called an *early-late tracking loop*. The idea behind the DLL is to correlate the input signal with three replicas of the code as seen in Fig. 8.

First step in Fig. 8: Convert the C/A code to baseband, by multiplying the incoming signal with a perfectly aligned local replica of the carrier wave. Afterwards the signal is multiplied with three code replicas. The three replicas are nominally generated with a spacing of $\pm\frac{1}{2}$ chip. After this second multiplication, the three outputs are integrated and dumped. The output of these integrations is a numerical value indicating how much the specific code replica correlates with the code in the incoming signal.

The three correlation outputs I_E, I_P, and I_L are then compared to see which one provides the highest correlation. Figure 7 shows an example of code tracking. In Fig. 7(a) the late code has the highest correlation, so the code phase must be decreased. In Fig. 7(b) the highest peak is located at the prompt replica, and the early and late replicas have equal correlation. In this case, the code phase is properly tracked.

The DLL with three correlators as in Fig. 8 is optimal when the local carrier wave is locked in phase and frequency. But when there is a phase error on the local carrier wave, the signal will be more noisy making it more difficult for the DLL to keep lock on the code. So instead the DLL in a GPS receiver is often designed as in Fig. 9.

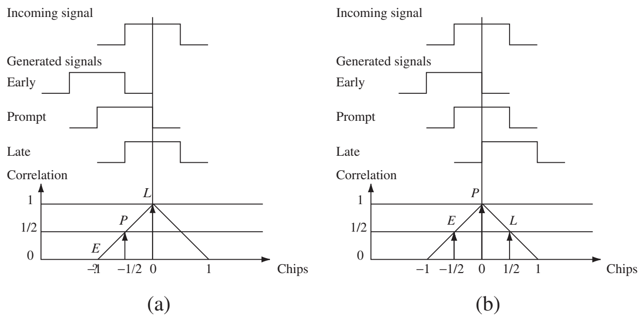

Fig. 7. Code tracking. Three local codes are generated and correlated with the incoming signal. (a) The late replica has the highest correlation so the code phase must be decreased, i.e., the code sequence must be delayed. (b) The prompt code has the highest correlation and the early and late have similar correlation. The loop is perfectly tuned in.

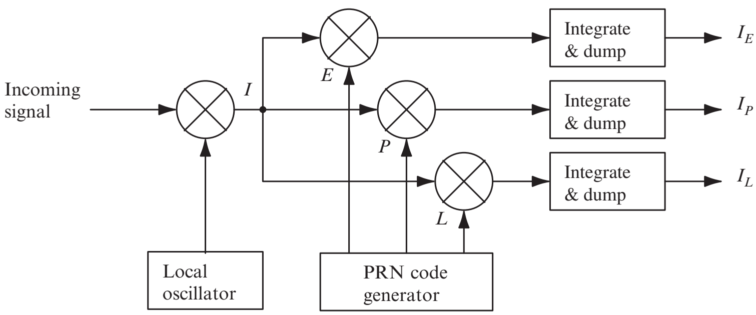

Fig. 8. Basic code tracking loop block diagram.

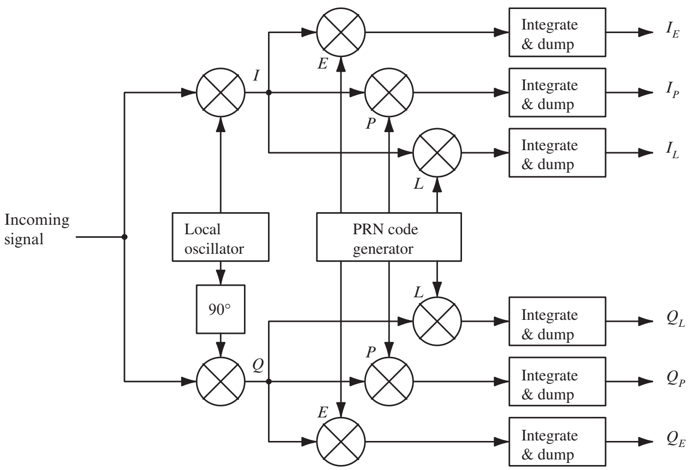

Fig. 9. DLL block diagram with six correlators.

The design in Fig. 9 has the advantage that it is independent of the phase on the local carrier wave. If the local carrier wave is in phase with the input signal, all the energy will be in the in-phase arm. But if the local carrier phase drifts compared to the input signal, the energy will switch between the in-phase and the quadrature arm. For demonstration purposes Fig. 10 shows such situation where the phase of the carrier replica drifts compared to the phase of the incoming signal. The upper plot shows the output of the three correlators in the in-phase arm and the lower plot shows the correlation output in the quadrature arm of the DLL with six correlators. This situation is a result of different frequencies for the signal and the replica; it results in a constantly changing phase difference (miss-alignment). There are a few reasons why this can happen, for example the PLL could be not in a lock state.

Figure 11 shows a case when the PLL is in a locked state. Because of the precise carrier replica from the PLL it is seen in Fig. 11 that the correlators are constant over time. This would not be the case if the carrier replica is not adjusted to match the frequency and phase of the incoming signal.

If the code tracking loop performance has to be independent of the performance of the phase lock loop, the tracking loop has to use both the in-phase and quadrature arms to track the code.

The DLL now needs a feedback to the PRN code generators if the code phase has to be adjusted. Some common DLL discriminators used for feedback are listed in Table 1.

The table shows one coherent and three non-coherent discriminators. The requirements of a DLL discriminator is dependent on the type of application and the noise in the signal. The discriminator function responses are shown in Fig. 12.

Figure 12 shows the coherent discriminator and three non-coherent discriminators using a standard correlator. The figure is produced from ideal ACFs and the space between the early, prompt, and late is $\pm\frac{1}{2}$ chip. The space between the

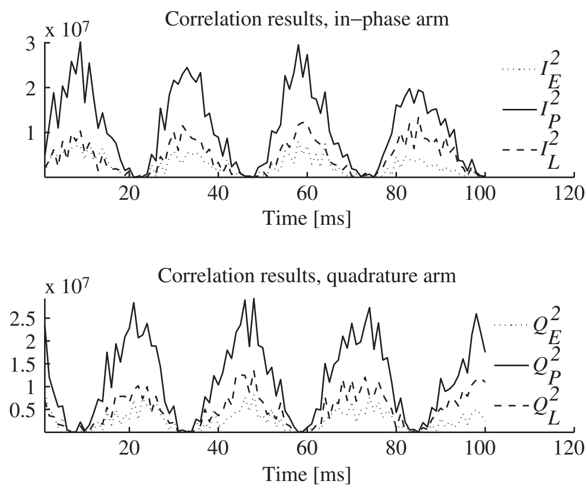

Fig. 10. Output of the six correlators in the in-phase and quadrature arms of the tracking loop. Acquisition frequency offset is 20 Hz and PLL noise bandwidth is 15 Hz (for demonstration purpose).

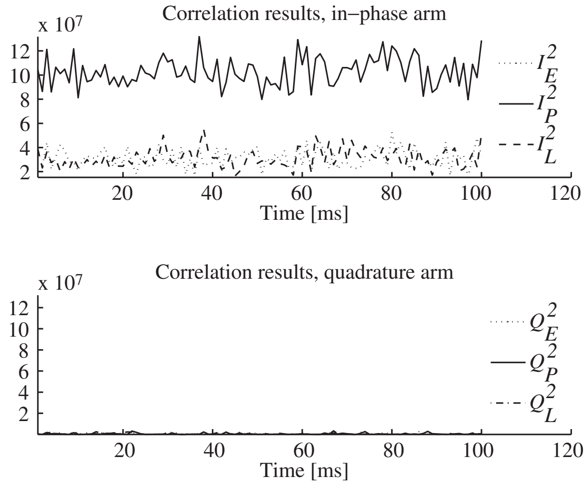

Fig. 11. Output of the six correlators in the in-phase and quadrature arms of the tracking loop. The local carrier wave is in phase with the input signal.

Table 1. Various types of delay lock loop discriminators and a description of them.

Type	Discriminator	Characteristics
Coherent	$D = I_E - I_L$	Simplest of all discriminators. Does not require the Q branch but requires a good carrier tracking loop for optimal functionality.
	$D = (I_E^2 + Q_E^2) - (I_L^2 + Q_L^2)$	Early minus late power. The discriminator response is nearly the same as the coherent discriminator inside $\pm\frac{1}{2}$ chip.
Non-coherent	$D = \dfrac{(I_E^2 + Q_E^2) - (I_L^2 + Q_L^2)}{(I_E^2 + Q_E^2) + (I_L^2 + Q_L^2)}$	Normalized early minus late power. The discriminator has a great property when the chip error is larger than a $\frac{1}{2}$ chip, this will help the DLL to keep track in noisy signals.
	$D = I_P(I_E - I_L) + Q_P(Q_E - Q_L)$	Dot product. This is the only DLL discriminator that uses all six correlator outputs.

early, prompt, and late codes determines the noise bandwidth in the delay lock loop. If the discriminator spacing is larger than $\frac{1}{2}$ chip, the DLL would be able to handle wider dynamics and be more noise robust, on the other hand a DLL with a smaller spacing would be more precise. In a modern GPS receiver the discriminator spacing can be adjusted while the receiver is tracking the signal. The advantage from this is that if the signal to noise ratio suddenly decreases, the receiver uses a wider spacing in the correlators to be able to handle a more noisy signal, and hereby a possible code lock loss could be avoided.

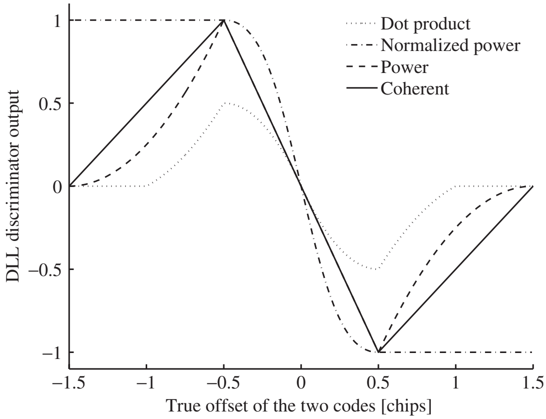

Fig. 12. Comparison between the common DLL discriminator responses.

The implemented tracking loop discriminator is the normalized early minus late power. This discriminator is described as

$$D = \frac{(I_E^2 + Q_E^2) - (I_L^2 + Q_L^2)}{(I_E^2 + Q_E^2) + (I_L^2 + Q_L^2)} \tag{10}$$

where I_E, Q_E, I_L, and Q_L are output from four of the six correlators shown in Fig. 10. The normalized early minus late power discriminator is chosen because it is independent of the performance of the PLL as it uses both the in-phase and quadrature arms. The normalization of the discriminator causes that the discriminator can be used with signals with different signal to noise ratios and different signal strengths.

The tracking loop generates three local code replicas. In this section, the chip space between the early and prompt replicas is half a chip.

As was described, the DLL can be modeled as a linear PLL and thus the performance of the loop can be predicted based on this model. In other words the loop filter design is the same, just parameter values are different.

5 Navigation Data Extraction

When the signals are properly tracked, the C/A code and carrier wave can be removed from the signal, leaving the navigation data bits. The value of a data bit is found by integration over a navigation bit period of 20 ms:

– Find start of subframe
– Decode the ephemeris data.

6 Estimation of Pseudorange

– Find common start for all satellites of a subframe. The accuracy of the pseudorange with a time resolution of 1 ms is 300 km

– The code tracking loop tells the precise start of the C/A code. Pseudorange accuracy of 8 meters. This value depends on signal sampling frequency.

Finally the receiver position is computed from the estimated pseudoranges. The next subsection describes a standard method for this computation.

7 Computation of Receiver Position

The most commonly used algorithm for position computations from pseudoranges is based on the ***least-squares method.*** This method is used when there are more observations than unknowns. This section describes how the least-squares method is used to find the receiver position from pseudoranges to four (Fig. 13) or more satellites.

Let the geometrical range between satellite k and receiver i be denoted ρ_i^k, and let c denote the speed of light. Let dt_i and dt^k be the receiver clock and satellite clock offsets. Let T_i^k be the tropospheric delay, I_i^k be the ionospheric delay, and e_i^k be the observational error of the pseudorange. Then the basic observation equation for the pseudorange P_i^k is

$$P_i^k = \rho_i^k + c(dt_i - dt^k) + T_i^k + I_i^k + e_i^k. \qquad (11)$$

The geometrical range ρ_i^k between the satellite and the receiver is computed as

$$\rho_i^k = \sqrt{(X^k - X_i)^2 + (Y^k - Y_i)^2 + (Z^k - Z_i)^2}. \qquad (12)$$

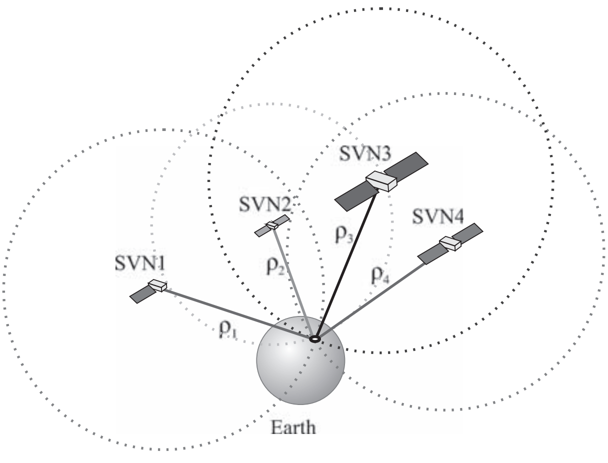

Fig. 13. The basic principle of GNSS positioning. With known position of four satellites SVNi and signal travel distance ρ_i, the user position can be computed.

Inserting (11) into (12) yields

$$P_i^k = \sqrt{(X^k - X_i)^2 + (Y^k - Y_i)^2 + (Z^k - Z_i)^2} + c(dt_i - dt^k) + T_i^k + I_i^k + e_i^k. \quad (13)$$

From the ephemerides—which include information on the satellite clock offset dt^k—the position of the satellite (X^k, Y^k, Z^k) can be computed. (The *M*-file **satpos** does the job.)

The tropospheric delay T_i^k is computed from an a priori model which is coded as **tropo**; the ionospheric delay I_i^k may be estimated from another a priori model, the coefficients of which are part of the broadcast ephemerides. There are four unknowns in the equation: X_i, Y_i, Z_i, and dt_i; the error term e_i^k is minimized by using the method of least squares. To compute the position of the receiver at least four pseudoranges are needed.

Equation (13) is nonlinear with respect to the receiver position (X_i, Y_i, Z_i), so the equation has to be ***linearized*** before using the least-squares method. We analyze the nonlinear range term in (13):

$$f(X_i, Y_i, Z_i) = \sqrt{(X^k - X_i)^2 + (Y^k - Y_i)^2 + (Z^k - Z_i)^2}. \quad (14)$$

Linearization starts by finding an initial position for the receiver: ($X_{i,0}$, $Y_{i,0}$, $Z_{i,0}$). This is often chosen as the center of the Earth (0,0,0).

The increments ΔX, ΔY, ΔZ are defined as

$$\begin{aligned} X_{i,1} &= X_{i,0} + \Delta X_i \\ Y_{i,1} &= Y_{i,0} + \Delta Y_i \\ Z_{i,1} &= Z_{i,0} + \Delta Z_i. \end{aligned} \quad (15)$$

These increments update the approximate receiver coordinates. So the Taylor expansion of $f(X_{i,0} + \Delta X_i, Y_{i,0} + \Delta Y_i, Z_{i,0} + \Delta Z_i)$ is

$$\begin{aligned} f(X_{i,1}, Y_{i,1}, Z_{i,1}) = f(X_{i,0}, Y_{i,0}, Z_{i,0}) &+ \frac{\partial f(X_{i,0}, Y_{i,0}, Z_{i,0})}{\partial X_{i,0}} \Delta X_i \\ &+ \frac{\partial f(X_{i,0}, Y_{i,0}, Z_{i,0})}{\partial Y_{i,0}} \Delta Y_i + \frac{\partial (X_{i,0}, Y_{i,0}, Z_{i,0})}{\partial Z_{i,0}} \Delta Z_i. \end{aligned} \quad (16)$$

Equation (16) only includes first order terms; hence the updated function *f* determines an approximate position. The partial derivatives in equation (16) come from (14):

$$\begin{aligned} \frac{\partial f(X_{i,0}, Y_{i,0}, Z_{i,0})}{\partial X_{i,0}} &= -\frac{X^k - X_{i,0}}{\rho_i^k} \\ \frac{\partial f(X_{i,0}, Y_{i,0}, Z_{i,0})}{\partial Y_{i,0}} &= -\frac{Y^k - Y_{i,0}}{\rho_i^k} \\ \frac{\partial f(X_{i,0}, Y_{i,0}, Z_{i,0})}{\partial Z_{i,0}} &= -\frac{Z^k - Z_{i,0}}{\rho_i^k}. \end{aligned}$$

Let $\rho^k_{i,0}$ be the range computed from the approximate receiver position; the first order linearized observation equation becomes

$$P^k_i = \rho^k_{i,0} - \frac{X^k - X_{i,0}}{\rho^k_{i,0}} \Delta X_i - \frac{Y^k - Y_{i,0}}{\rho^k_{i,0}} \Delta Y_i - \frac{Z^k - Z_{i,0}}{\rho^k_{i,0}} \Delta Z_i + c(dt_i - dt^k) + T^k_i + I^k_i + e^k_i \tag{17}$$

where we explicitly have

$$\rho^k_{i,0} = \sqrt{(X^k - X_{i,0})^2 + (Y^k - Y_{i,0})^2 + (Z^k - Z_{i,0})^2}. \tag{18}$$

A least-squares problem is given as a system $A\boldsymbol{x} = \boldsymbol{b}$ with no exact solution. A has m rows and n columns, with $m > n$; there are more observations $b_1, \ldots, b_m$ than free parameters $x_1, \ldots, x_n$. The best choice, we will call it $\hat{\boldsymbol{x}}$, is the one that minimizes the length of the error vector $\hat{\boldsymbol{e}} = \boldsymbol{b} = A\hat{\boldsymbol{x}}$. If we measure this length in the usual way, so that $\|\boldsymbol{e}\|^2 = (\boldsymbol{b} - A\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{b} - A\boldsymbol{x})$ is the sum of squares of the m separate errors, minimizing this quadratic gives the normal equations

$$A^T A\hat{\boldsymbol{x}} = A^T \boldsymbol{b} \quad or \quad \hat{\boldsymbol{x}} = (A^T A)^{-1} A^T \boldsymbol{b}. \tag{19}$$

The error vector is

$$\hat{\boldsymbol{e}} = \boldsymbol{b} - A\hat{\boldsymbol{x}}. \tag{20}$$

The covariance matrix for the parameters $\hat{\boldsymbol{x}}$is

$$\Sigma_{\hat{x}} = \hat{\sigma}^2_0 (A^T A)^{-1} \quad with \quad \hat{\sigma}^2_0 = \frac{\hat{\boldsymbol{e}}^{\mathrm{T}} \hat{\boldsymbol{e}}}{m-n}. \tag{21}$$

The linearized observation equation (17) can be rewritten in a vector formulation

$$P^k_i = \rho^k_{i,0} + \begin{bmatrix} -\frac{X^k - X_{i,0}}{\rho^k_{i,0}} & -\frac{Y^k - Y_{i,0}}{\rho^k_{i,0}} & -\frac{Z^k - Z_{i,0}}{\rho^k_{i,0}} \end{bmatrix} \begin{bmatrix} \Delta X_i \\ \Delta Y_i \\ \Delta Z_i \\ c\,dt_i \end{bmatrix} - c\,dt^k + T^k_i + I^k_i + e^k_i. \tag{22}$$

We rearrange this to resemble the usual formulation of a least-squares problem $A\boldsymbol{x} = \boldsymbol{b}$

$$\begin{bmatrix} -\frac{X^k - X_{i,0}}{\rho^k_{i,0}} & -\frac{Y^k - Y_{i,0}}{\rho^k_{i,0}} & -\frac{Z^k - Z_{i,0}}{\rho^k_{i,0}} & 1 \end{bmatrix} \begin{bmatrix} \Delta X_i \\ \Delta Y_i \\ \Delta Z_i \\ c\,dt_i \end{bmatrix} = P^k_i - \rho^k_{i,0} + c\,dt^k - T^k_i - I^k_i - e^k_i. \tag{23}$$

A unique least-squares solution cannot be found until there are $m \geq 4$ equations. Let $b^k_i = P^k_i - \rho^k_{i,0} + c\,dt^k - T^k_i - I^k_i - e^k_i$ and the final solution comes from

$$
A\boldsymbol{x} = \begin{bmatrix} -\frac{X^1 - X_{i,0}}{\rho_{i,0}^1} & -\frac{Y^1 - Y_{i,0}}{\rho_{i,0}^1} & -\frac{Z^1 - Z_{i,0}}{\rho_{i,0}^1} & 1 \\ -\frac{X^2 - X_{i,0}}{\rho_{i,0}^2} & -\frac{Y^2 - Y_{i,0}}{\rho_{i,0}^2} & -\frac{Z^2 - Z_{i,0}}{\rho_{i,0}^2} & 1 \\ -\frac{X^3 - X_{i,0}}{\rho_{i,0}^3} & -\frac{Y^3 - Y_{i,0}}{\rho_{i,0}^3} & -\frac{Z^3 - Z_{i,0}}{\rho_{i,0}^3} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -\frac{X^m - X_{i,0}}{\rho_{i,0}^k} & -\frac{Y^m - Y_{i,0}}{\rho_{i,0}^m} & -\frac{Z^m - Z_{i,0}}{\rho_{i,0}^m} & 1 \end{bmatrix} \begin{bmatrix} \Delta X_{i,1} \\ \Delta Y_{i,1} \\ \Delta Z_{i,1} \\ c\,dt_{i,1} \end{bmatrix} = \boldsymbol{b} - \boldsymbol{e}. \tag{24}
$$

The solution $\Delta X_{i,1}$, $\Delta Y_{i,1}$, $\Delta Z_{i,1}$, is added to the approximate receiver position to get the next approximate position:

$$
\begin{aligned} X_{i,1} &= X_{i,0} + \Delta X_{i,1} \\ Y_{i,1} &= Y_{i,0} + \Delta Y_{i,1} \\ Z_{i,1} &= Z_{i,0} + \Delta Z_{i,1}. \end{aligned} \tag{25}
$$

The next ***iteration*** restarts from (22) to (25) with $_{i,0}$ replaced by $_{i,1}$. These iterations continue until the solution $\Delta X_{i,1}$, $\Delta Y_{i,1}$, $\Delta Z_{i,1}$ is at meter level. Often 2–3 iterations are sufficient to obtain that goal, are discussed in [2].

The present description of the software-defined GPS receiver is based on [3]. Further developments in the project can be followed at **gps.aau.dk/softgps.**

References

[1] Anonymous, "An introduction to direct-sequence spread-spectrum communications," 2003, http://www.maxim-ic.com/appnotes.cfm/appnote_number/1890.

[2] G. Strang and K. Borre, *Linear Algebra, Geodesy, and GPS*. Wellesley, MA: Wellesley-Cambridge Press, 1997.

[3] K. Borre, D. Akos, N. Bertelsen, P. Rinder, and S. H. Jensen, *A Software-Defined GPS and Galileo Receiver: Single-Frequency Approach*. Boston Basel Berlin: Birkhäuser, 2006.