Decoding GPS satellite signals received by terrestrial digital video-broadcast (DVB-T) receivers

J.-M Friedt, G. Cabodevila, June 16, 2015

Positioning by processing signals emitted by satellites – GNSS or Global Navigation Satellite Service – has become so ubiquitous that one no longer even thinks of the technical challenges required by this fascinating technology. Based on a constellation of satellites, each embedding several atomic clocks, precisely time-stamped signals allow the positioning of the receiver by triangulation. Thanks to a dedicated knowledge of the communication protocol, we envision more interesting applications than simple positioning: we will here consider, after acquiring raw radiofrequency signals with a receiver initially designed for recording Terrestrial Digital Video Broadcast (DVB-T), recovering the basic information transmitted by the satellites. We will see that several feedback loops are needed to recover the carrier frequency and the navigation messages, and will tackle the issue of extracting signals from a given satellite when all vehicles of the constellation are emitting on the same carrier frequency (1575.42 MHz). Recovering meaningful navigation bits demonstrates that our approach is sound.

We have already discussed multiple times the use of terrestrial digital video broadcast receivers for listening to radiofrequency signals, processed through the GNURadio software for extracting information stored in the carrier modulation [1]. We have gone all the way to demonstrating the reception of signals emitted by low-Earth orbiting satellites, at an altitude of a few hundreds of kilometers, encoded by classical amplitude and frequency modulation schemes. GPS signals used to have been recorded thanks to the `gnss-sdr` software [1], free software yet too complex to understand the signal processing steps involved when processing the satellite signals by reading source codes. In this article, the experimental setup is most basic since it includes a terrestrial digital video broadcast (television) receiver compatible with `rtl-sdr` [2] – hence an E4000 or more probably a R820T(2) receiver followed by a Realtek RTL2832U analog to digital converter – a GPS active antenna as for example sold by Lextronic [3] for 12 euros, and a bias T made of a capacitor to prevent the DC component from reaching the radiofrequency receiver input (a few hundreds of picofarads) and an inductor (a few microhenrys) to prevent the radiofrequency signal from reaching the antenna power supply provided by the USB port. Notice that `gnss-sdr` will only work with an Elonics E4000 front-end, now discontinued, and applying these processing steps to a Rafael Micro R820T is new, most significantly with the latter being plagued by a huge frequency offset, much larger than those found on the E4000 whose accuracy was already poor. Hence, the *financial investment to reproduce these experiments is of the order of about 30 euros.* We will see that decoding GPS signals means recording a radiofrequency signal (lasting a few seconds at most), and cross-correlating with local copies of all possible code patterns. Since recovering a phase modulated information is only possible once the carrier has been canceled by the local oscillator. Searching for the presence of a GPS signal in a radiofrequency signal sample requires a heavy computational load in which we sweep all possible satellite identification codes, and for each code we sweep the local oscillator frequency over all possible Doppler shifts.

1 Introduction

We wish to push one step further the demonstration of how powerful software defined radio processing is and how flexible the Terrestrial Digital Video Broadcast (DVB-T) receiver is by analyzing signals in the radiofrequency band used by GPS [2, 3]. GPS – Global Positioning System – is one of the satellite constellation designed for locating the receiver by triangulation, and is part of the broader field of GNSS – Global Navigation Satellite Service – also implemented by Russia with GLONASS (using a very different modulation scheme which will not be described here), might one day become available through the European Galileo constellation, by China with Beidou and, less advanced, by India with IRNSS. Positioning of the receiver is the result of an accurate triangulation of the signals emitted by

[1] `gnss-sdr.org`

[2] `sdr.osmocom.org/trac/wiki/rtl-sdr`

[3] `http://www.lextronic.fr/P847-antenne-gps-amplifiee-magnetique.html`

several satellites whose position in space is assumed to be very well known, and whose time of flight from the emitter to the receiver is precisely computed. Remembering that electromagnetic waves propagate at a velocity of 300 m/μs, then locating the receiver with a 3 m resolution requires the computation of the time of flight with a 10 ns resolution after a total time of flight of over 70 ms. Recovering such an accurate timing signal allows for recovering much more interesting information than position [6]: examples include electron density in the ionosphere, moisture concentration in the atmosphere, or detecting pressure waves propagating in air following an earthquake. Signals received on the ground have also been used for RADAR [4] applications in which target reflector properties are deduced by analyzing the reflected signal with respect to the one received directly from the satellite : moisture level (through the reflection coefficient) or distance to the target. Finally, reproducing on the ground a clock whose stability properties combine the stability of airborne atomic clocks provides the perspective of accurate time transfer between multiple sites located far apart one from the other [4, 5], a mandatory requirement for interferometric event localization (radio-telescope, lightning strike positioning for example [5]).

Although the constellation of satellites emitting signals whose frequency is very stable over time is commonly called GPS, the embedded clocks have been the subject of several evolutions in order to improve performance and reliability: initially fitted solely with rubidium clocks, which are secondary time references, a fourth cesium clock (primary reference since the second is defined through a relationship with the transition frequency between two states of cesium atoms [7]) has been added once a space-compatible version became functional, to be finally removed since the expected improvements were not met [8]. In the following application, we shall focus on a solely software processing solution, providing the flexibility needed for fast prototyping and adapt to the various applications mentioned earlier: the software defined radio (SDR) approach will once again demonstrate how powerful it is.

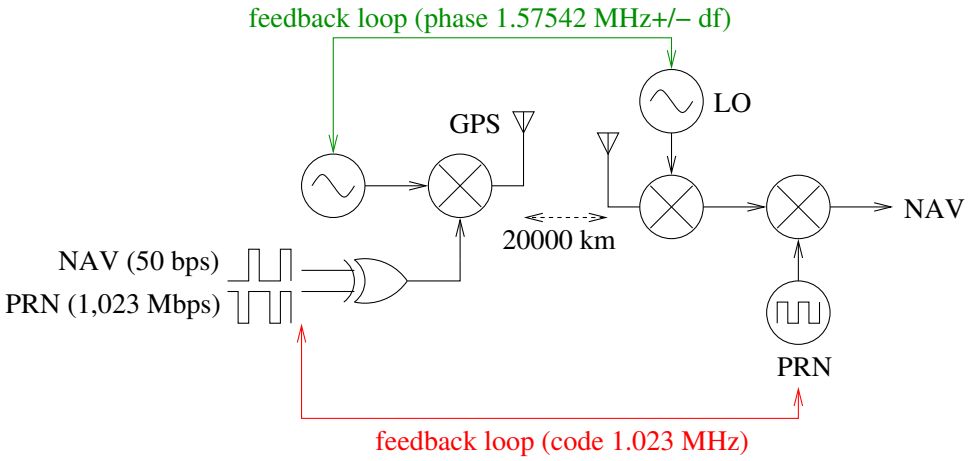


Figure 1: Modulation scheme of the information emitted by a satellite of the GPS constellation (left) when transmitting towards the ground (right). Our objective on the ground is to recover the time of flight of the signal between the satellite and the receiver, and decode the navigation message carried by the signal.

Processing signal recovered from the GPS satellite constellation requires answering several questions:

1. satellites are moving since they are located on an orbit at an altitude between 20000 and 21000 km. Although the embedded atomic clocks exhibit utmost stability, the motion of the satellite induces a shift of the carrier frequency shift with respect to its nominal value which must be compensated for on the receiver side. Additionally, the oscillator clocking the DVB-T receiver is of very poor quality: the carrier frequency identification algorithm must be able to compensate for fluctuations of the local copy of the carrier frequency due to temperature fluctuations (random shift) and the

[4] A virtual visit of the RADAR Museum is accessible at `http://www.musee-radar.fr/360-generartion-radar-2014/radar.html`

[5] `http://www.zeus.iag.usp.br/system.php?lan=en`

bias to the targeted frequency (offset) due to the poor assembly quality of the oscillator clocking the DVB-T receiver.

2. *all the satellites communicate of the same carrier frequency*: a strategy for identifying which message belongs to which satellite must be implemented, despite all the other messages emitted by the other satellites being broadcast at the same time. Sharing the spectrum by encoding differently the messages sent by each satellite is called CDMA – *Code Division Multiple Access* – a concept widely used in digital communications.

3. Having identified which information is emitted by which satellite and what the carrier frequency offset is, we must finally decode the message encoded in the phase of the signal 1). The mixer on the emission side induces (bit equal to 1) or does not induce (bit equal to 0) a phase rotation of 180^o of the carrier: the modulation scheme is BPSK [1] (Binary Phase Shift Keying) in which the phase of the signal transmits an exclusive-ORed combination of identification (code) and navigation (useful message) bits.

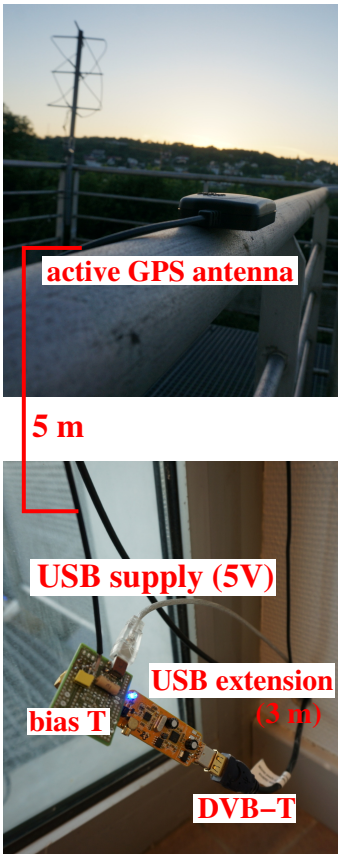
A general rule is that for N variables needed to transmit a signal through a radiofrequency link, N feedback loops will be needed to recover each parameter of the emitter. Selecting various carrier frequencies when transmitting signal over a radiofrequency link – and hence shifting the spectrum with respect to baseband (centered around 0 Hz) – answers the requirement of sharing the radiofrequency spectrum and shrinking the dimensions of the antenna needed to convert the electrical signal to an electromagnetic wave. Here we must recover at least two parameters in order to decode the information carried by the radiofrequency wave: the carrier frequency, and the rate at which the transmitted message is sent. Indeed, there is no reason to believe that the oscillators on both ends of the link behave identically, and identifying the carrier frequency provides the means to synchronize the two oscillators clocking the emitter and the receiver. In the case of GPS, we must find the solution for $N = 3$ free parameters: the carrier frequency (between the signal emitted by the satellite and the receiver on the ground), the code identifying the emitting satellite, and the frequency (or phase, since the frequency is the derivate of the phase) at which the code is transmitted. We will see later that the main challenge lies in the modulation scheme in which the phase is used for two purposes: controlling the local oscillator in order to track the received signal carrier frequency, and decoding the navigation message emitted by the satellite.

As a conclusion, the problems we will have to solve are summarized below:

1. all satellites of the constellation emit on the same nominal frequency (unlike the Russian GLONASS constellation). The radiofrequency receiver hence records a signal representative of the sum of the contributions of all the satellites visible by the antenna, from which each single satellite component must be extracted. Each satellite is identified by a unique code: such a scheme is called CDMA (Fig. 2),

2. the signal received on the ground is below the thermal noise, considering the sampling bandwidth needed to process GPS data. Indeed, any electronic component is plagued by an intrinsic noise source due to thermal motion of the electrons carying the electrical information: this wideband noise is integrated over the measurement bandwidth. The wider the measurement bandwidth, the higher this noise contribution is. As usually found in thermodynamics, the relationship between temperature and noise spectral density is given by the Boltzmann constant, $k_B = 1,38 \times 10^{-23}$ J/K, which yields when going to a logarithmic scale to $10 \times \log_{10}(k_B \cdot T) = -204$ dBW (=-174 dBm) at ambient temperature ($T = 293$ K). This aspect is unsettling at first, meaning that it is not possible to "simply" monitor a GPS signal on a spectrum analyzer, but we will see how analyzing a long (1 ms) sequence of samples containing a known pattern will allow to accumulate coherently the power of the signal emitted by the satellite and hence improve the signal power while decreasing the thermal noise contribution until a positive signal to noise ratio is reached.

3. finally, all these processing steps require extracting an information from the radiofrequency received from the satellite, and doing so requires removing the carrier frequency. However, the satellite is moving along its orbit and hence induces a time-varying frequency shift: we must identify and cancel this frequency offset in order to decode the information included in the phase modulation.



Experimental setup

The design of the bias T has already been discussed in [1]: a capacitor C of a few hundreds of pF yields an impedance $1/(C\omega)$ of about an ohm at an angular frequency $\omega = 2\pi \times 1.5742$ GHz while efficiently blocking the continuous component of the supply voltage to the antenna, and an inductor L of about 10 μH induces and impedance $L\omega$ of about 100 kΩ, well above the capacitor impedance and hence efficiently blocking the propagation of the radiofrequency signal in the antenna power supply.

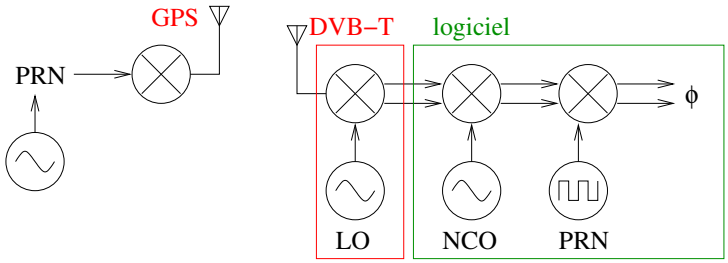


Figure 2: CDMA communication basics: all satellites emit on the same frequency, but using a message encoded with different pseudo-random sequences (PRN).

Past sensitivity measurements of DVB-T receivers [1] showed that the power from an incoming amplitude or frequency modulated radiofrequency signal around 1.5 GHz needed to reach a 10 dB signal to noise ratio after demodulation is -95 dBm. Two questions hence arise: how does the power of a signal

emitted from a satellite located at an altitude of 20000 km compare with this limit at ground level, and how do these two values compare with the thermal noise integrated over the 2 MHz sampling bandwidth.

GPS satellites emit a signal with a power P_E of 25 to 50 W, or +14 to 17 dBW [9], with an antenna exhibiting a gain of $G_E = 13$ dBi. Heaving spread this power on the sphere surrounding the satellite at an altitude of 21000 km – the maximum distance between the emitter (satellite) and the receiver (us, at ground level) – only a minute fraction of this power is recovered with losses of 182 dB (*Free Space Propagation Loss* induces an attenuation by $\left(\frac{4\pi \cdot d}{\lambda}\right)^2$ by spreading uniformly the incoming power over a sphere of area $4\pi \cdot d^2$ at a distance d from the emitter, with λ the transmitted signal wavelength, 19 cm at 1.57542 GHz), and the received power on the ground is $P_R = -155$ dBW or -125 dBm for an isotropic receiving antenna and hence a unitary gain G_R (Fig. 3).

How does this received power compare with thermal power over a $B = 2$ MHz bandwidth ? The thermal noise density for an antenna facing an area of temperature T is $k_B \cdot T \cdot B$ (with k_B the Boltzmann constant), or here -174 dBm/Hz+10·log$_{10}$(2 MHz)=-111 dBm. Thermal noise is hence 14 dB above the GPS signal reaching ground, so will we be able to extract any useful information ?

The signal transferred on the GPS carrier follows a known pattern distributed over the whole measurement bandwidth. We will obviously not claim to fail to comply with Shannon's channel capacity theorem which tells us that the communication bandwidth, C in bits/s, is related to the channel bandwidth B in Hz and to the signal to noise ratio SNR (no unit) of the link by $C = B \cdot \log_2(1 + SNR)$: increasing the number of speakers requires increasing the bandwidth, here by adding to the transmitted message (the bits of the navigation message) an identifier unique to each satellite, transmitted much faster than the message itself. Searching for this identification pattern will allow extracting the meaningful information from the noise.

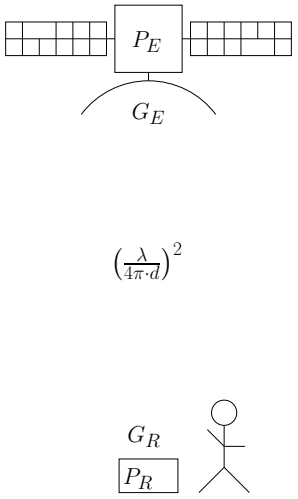


Figure 3: Link budget between the satellite and ground.

The 10-bit pseudo-random code (see inset 1 for a detailed description of this aspect), generated at a rate of 1.023 MHz, repeats every $2^{10} - 1 = 1023$ values, or 1 kHz. The signal is hence coherently integrated over a duration of 1 ms, yielding a compression gain [6] of $1.023 \cdot 10^6 \times 10^{-3} = 1023$ or 30 dB. Integrating over 10 successive periods of the targeted code increases this value to 40 dB. Hence, the signal to noise ration after extracting the valuable information is SNR=$(-125 + 30) - (-111) = 16$ dB. The signal will hence be clearly visible above noise.

From another perspective, are such signals above the noise floor of the receiver (Fig. 4) ? The active antenna most GPS receivers are fitted with include a low noise amplifier with +27 dB gain. The signal received by the DVB-T dongle is thus -125+27=-98 dBm. This value is dangerously close to the -95 dBm noise floor level we have measured [1], but this noise floor was obtained in order to yield a 10 B signal to noise ration after AM or FM demodulation. We will thus be able to use the resulting signal.

We notice a property which will provide the core justification for later analysis: the phase of the cross correlation conveys the phase encoding on the original signal. The demonstration is simple by considering that the cross-correlation of a signal with a known pattern is a linear process: if we name $x(\tau)$ the cross-correlation of the acquired signal s with the code c free of phase encoding, then

$$x(\tau) = \int_{-\infty}^{+\infty} s(t) \cdot c^*(t - \tau) \cdot dt$$

(notice that the cross-correlation has been here defined for complex arguments for which the complex conjugate of one of the terms is needed – without the complex conjugate we will indeed have a cross-correlation phase dependent on the code phase, but with the wrong sign). If now we add a phase element to the copy c' of the code introduced in the signal, so that $c'(\tau) = c(t) \cdot \exp(j\varphi)$, then the cross-correlation becomes

$$x'(\tau) = \int_{-\infty}^{+\infty} s(t) \cdot c^*(t - \tau) \cdot \exp(j\varphi) \cdot dt = \exp(j\varphi) \cdot \int_{-\infty}^{+\infty} s(t) \cdot c^*(t - \tau) \cdot dt = \exp(j\varphi) \cdot x(\tau)$$

[6] the compression gain was discussed in [10]: it characterizes the capability of a correlator to coherently accumulate the energy of the searched pattern by which the signal is cross-correlated, while noise slowly averages down to 0 and does not benefit from coherent accumulation.

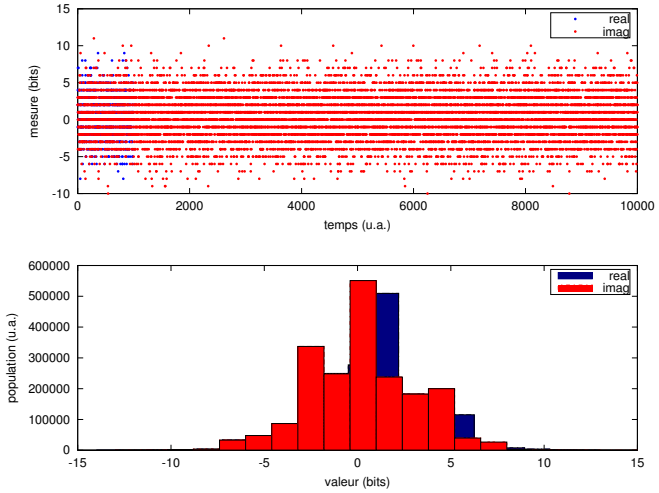


Figure 4: I and Q coefficients measured around 1.57542 GHz by a DVB-T receiver. Top is the time evolution, and bottom the histogram of the measurements. Only 3 (most values lie in the -4 to +4 range) of the available 8 bits are actually used.

so we indeed recover φ in the phase of the cross-correlation. Hence, if $c \to c{\cdot}\exp(j\varphi)$, then $x \to x{\cdot}\exp(j\varphi)$. The BPSK encoding introduced by the GPS emitted in the code is indeed recovered in the phase of the cross-correlation.

The sum (discrete version of the integral) of c with s coherently accumulates when s includes the pattern of c. This operation is repeated for every possible offsets τ of c in s: $xcorr(c,s)(\tau) = \int_{-\infty}^{+\infty} c(t) \cdot s(t-\tau) \cdot dt$. If the signal acquired by the radiofrequency receiver is the sum of multiple emitters, each encoded by its individual signature $c_k(t)$, then it is best if $xcorr(c_i, c_j) \simeq 0$ if $i \neq j$ and $xcorr(c_i, c_i) \simeq 1$. This way, only the cross-correlation of the signal $a(t)$ with the "right" code yields a non-null cross-correlation (Fig. 7).

In the particular case of GPS, the code $c(t)$ is generated by two imbricated pseudo-random generator (*Linear Feedback Shift Register* – LFSR). The detailed description of this code – named from now on PRN – is of little interest, and the only useful information is that 37 possible codes exist with the orthogonality property mentioned earlier, with only the first 31 being used at the moment.

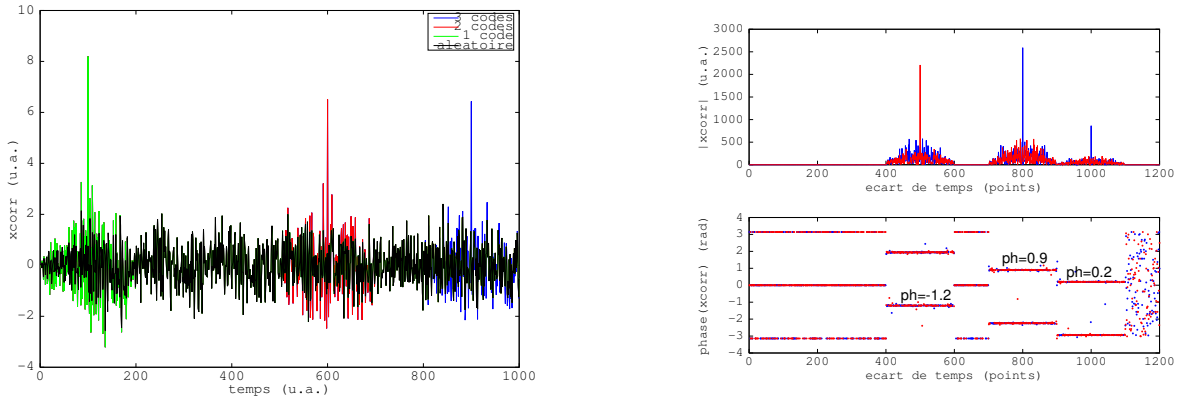
2 Code orthogonality

Orthogonality means that only the correlation of a code with a copy of this same bit sequence generates a non-null result. In the case of GPS, the pseudo-random pattern allowing for the identification of the satellite from which the signal has been sent is generated by a 10-bit long linear feedback shift register (LFSR) whose properties have been extensively described by Yann Guidon [11]. GPS extends the complexity by imbricating two LFSR. Rather than implement a new LFSR generation algorithm, we have used the pseudo-random generator script for Matlab (running with GNU/Octave) available at `http://www.mathworks.com/matlabcentral/fileexchange/14670-gps-c-a-code-generator`. However, we will use this software to experimentally assess on the one hand the orthogonality property of this pattern generator, but also how robust they are to sampling rate differences (or, equivalently, to the carrier modulated by these codes).

The basics of extracting the code modulated as a binary phase shift keying (BPSK) in order to recover which satellite has sent a signal on the common carrier frequency within the satellite constellation is based on cross-correlating the signal received by the radiofrequency receiver with each of the possible patterns used to identify each satellite. The code is sampled at 1.0123 MHz, and a 10-bit long LFSR repeats every $2^{10} - 1 = 1023$ bits. Hence, we will obtain, if the correct sequence is used, a cross-correlation maximum

The concept of encoding the source of an emitted signal is illustrated with a simple GNU/Octave simulation. Assume that only noise is emitted, `aleat=rand(1000,1);` with a mandatory null average value, `aleat=aleat-mean(aleat);` (we leave to the reader watching what happens for non-null average sources – remember that cross-correlating two constant functions yields a triangle). Assume now that a message is encoded as a *known* pseudo-random sequence `code=rand(100,1);code=code-mean(code);` added to various parts of the original signal: `aleat(1:100)=aleat(1:100)+code;` and `aleat(end-99:end)=aleat(end-99:end)+code;` We observe that the cross-correlation between the searched pattern and the received signal synthesized as described here exhibits maxima for each position offset of the wanted signal: `plot(xcorr(code,aleat))`. A most useful additional property of cross-correlation, being a linear transform, is that it also conveys the phase encoded in the signal. Hence, extending the previous example but now applied to complex signals conveying the information as a phase φ of the shape $\exp(j\varphi)$. Then

```
signal=rand(1000,1); signal=signal-mean(signal);
code=rand(100,1); code=code-mean(code);
signal(1:100)=signal(1:100)+100*code*exp(j*0.2);
signal(201:300)=signal(201:300)+300*code*exp(j*0.9);
code2=rand(100,1); code2=code2-mean(code2);
signal(501:600)=signal(501:600)+300*code2*exp(-j*1.2);
subplot(211);plot(abs(xcorr(code,signal)));hold on
plot(abs(xcorr(code2,signal)),'r');xlim([0 1200])
subplot(212);plot(angle(xcorr(code,conj(signal))),'.');hold on
plot(angle(xcorr(code2,conj(signal))),'r.');xlim([0 1200])
```



Left: magnitude of the cross-correlation as the code we are looking for has been added to a random signal for three different delays. Right: top is the magnitude of the cross-correlation, and bottom the phase of the cross-correlation which carries the information (`ph`) introduced in the code phase. This time the cross-correlation operates on complex values.

Table 1: Illustration of information encoding by a known pseudo-random pattern.

every time our local copy of the pattern matches that of the received signal, hence every 1 ms. On the other hand, the cross-correlation conveys the phase information which the modulator set at emission on the carrier: exploiting the phase of the cross correlation every time its magnitude is maximum will allow recovering the transmitted symbols. Separating which symbol was sent by which satellite then becomes a matter of running the various patterns of the codes identifying satellites visible in the sky at a given time.

Remember that cross-correlation is only rarely computed in time or space domain: most often it is more efficient to switch to the frequency domain (Fourier domain) to take advantage of the computational efficiency of the Fast Fourier Transform. Indeed, the Fourier transform of a convolution is the product of the Fourier transforms of the functions being considered, while time reversal to go from a convolution to a correlation is obtained by taking the complex conjugate of one of the two function Fourier transforms.

A subtlety remains in that the pseudo-random pattern modulates a carrier. However, there is no

reason for the DVB-T local oscillator frequency to exactly match the GPS satellite embedded atomic clock frequency. On the contrary, the frequency of the signal emitted by the satellite is shifted by the Doppler effect – a variable shift depending on the position of the satellite in the sky – and the local oscillator drifts with environmental conditions (temperature, stress [1]). Since we do not know in advance what these frequency shits add to, we must search for all possible values until a rough estimate is obtained, which will later be fine tuned until the difference between local and incoming carrier frequencies cancel by locking a feedback loop. We are not yet there: at the moment, in order to minimize computation time, we must assess how many frequency steps will be needed during this initial search. The wider the possible frequency difference is, the more processing steps will be needed for a given frequency step. On the other hand, if cross-correlating code patterns is robust to some carrier frequency difference, then the frequency steps might be increased in order to minimize computation time. La Fig. 5 illustrates the result of computing cross-correlations when one of the copies of the pattern is sampled 0.3 and 1% faster than the other. We see that the 1% error yields a drop and spreading of the cross-correlation peak – still visible in these ideal conditions but useless on real noisy signals – and that a frequency shift of more than a few fractions of a percent will yield a cross correlation peak hidden in noise. We have experimentally observed that searching with steps from 100 to 500 Hz yields acceptable results. Considering that the Doppler shift is ± 5 kHz for a fixed receiver [1], we still have to run 31×100=3100 cross-correlation computations to find all possible satellites visible in the sky (31 possible pseudo-random code patterns, and 100 frequency steps). This computation additionally assumes that the local oscillator is not biased, which is certainly not the case of the low cost DVB-T receivers we are using. We have observed *offsets of ± 100 kHz* (or ± 60 ppm at 1.5 GHz), yielding a significant extension of the initial search range and multiplying by a factor the computation duration. This last issue justifies for example current research on high accuracy compact oscillators (referenced to atomic transitions [12]) in order to reduce acquisition time (name commonly used to describe this processing step when a GPS receiver has just been switched on) of GPS signals.

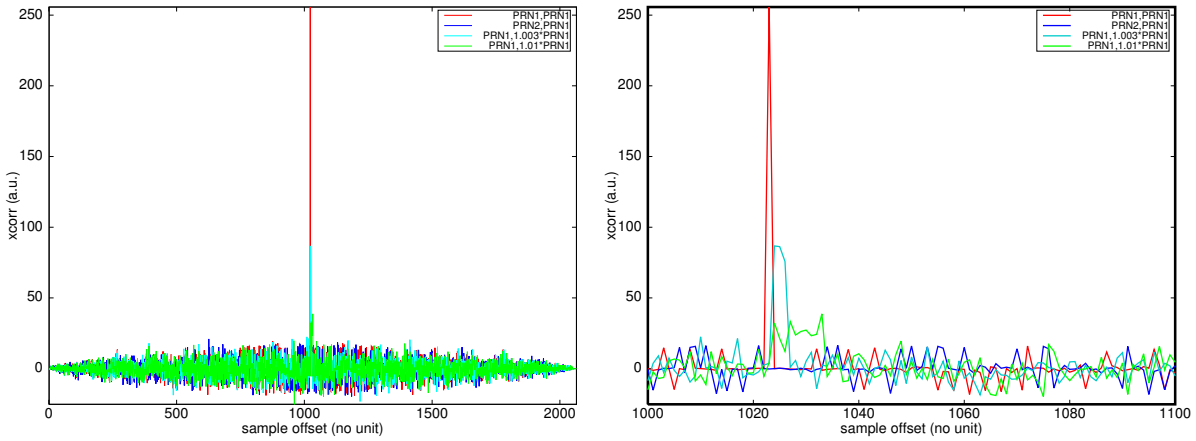


Figure 5: Left: in blue the cross-correlation of two different pseudo-random patterns (providing an estimate of the noise floor of this computation), and cross-correlation of the pattern with itself (red) or scaled versions by 1% (green) and 0.3% (cyan) of the time axis. Right: zoom on the cross-correlation peak for visualizing spreading and drop of the cross-correlation peak as the sampling rate difference becomes too large.

3 Demodulation principle

We have discussed in a previous article [1] the need to recover the carrier frequency in order to cancel it on the receiver and thus recover the information encoded in the recorded signal. As a reminder, a signal $s(t)$ is received by a radiofrequency receiver, including a phase modulation $\varphi(t)$ so that $s(t) = \sin(\omega_R \cdot t + \varphi(t))$, with ω_R the angular frequency of the emitted signal. Shifting the frequency by mixing the received signal with the local oscillator whose angular frequency is ω_L followed by a low pass filter generates a signal

digitized by the fast analog to digital converter as $S(t) = \sin((\omega_R - \omega_L) \cdot t + \varphi(t))$. We can only extract $\varphi(t)$ by canceling the phase rotation contribution $(\omega_R - \omega_L) \cdot t$, which means controlling ω_L to match ω_R. Indeed, ω_R varies due to the Doppler shift as the satellite moves along its orbit, while ω_L fluctuates due to temperature variations on the ground or stress if the receiver is moved. Finally, the classical approach is not only to consider mixing with $\sin(\omega_L \cdot t)$ which yields an estimate of $\cos(\varphi(t))$ when $\omega_R = \omega_L$, but also with $\sin(\omega_L \cdot t + \pi/2) = \cos(\omega_L \cdot t)$ in order to generate, during the mixing, $\sin(\varphi(t))$: these two components are called I and Q (In-phase and Quadrature), useful to compute the phase $\varphi(t) = \arctan(Q, I)$.

All this works very well on a single phase-modulated carrier: we had presented how a signal processing trick – the Costas loop – allows for removing the phase modulation and hence track ω_L on ω_R in order to extract $\varphi(t)$. As a reminder, in the particular case of the two-state BPSK, $\varphi(t) \in [0; \pi]$ et $2\varphi = 0[2\pi]$ so that squaring the signal cancels the phase modulation.

The problem is more tricky with CDMA in which the sum of the contributions of several emitters overlap and is acquired. In this case, we must identify which component belongs to which emitted first, and then track the carrier of each individual emitter. The solution to this detection scheme lies in the cross-correlation of the received signal with local copies of the codes identifying each emitter (Fig. 6).

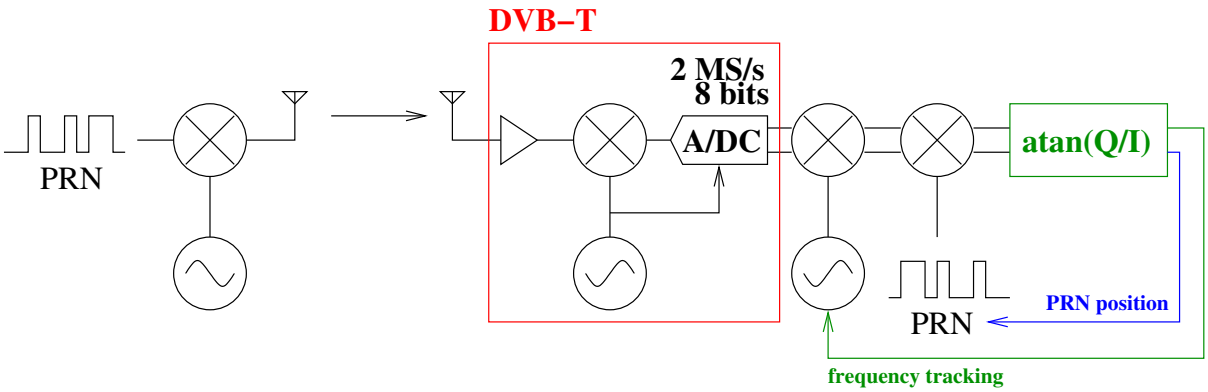


Figure 6: CDMA reception principle: we must on the one hand identify the carrier frequency of the received signal, but also the clock that was used to sequence the code, since both clocks (sampling on the ground and pseudo-random generator in space) have no reason to be synchronous, if only because of the poor quality of the oscillator clocking the ADC of the DVB-T receivers.

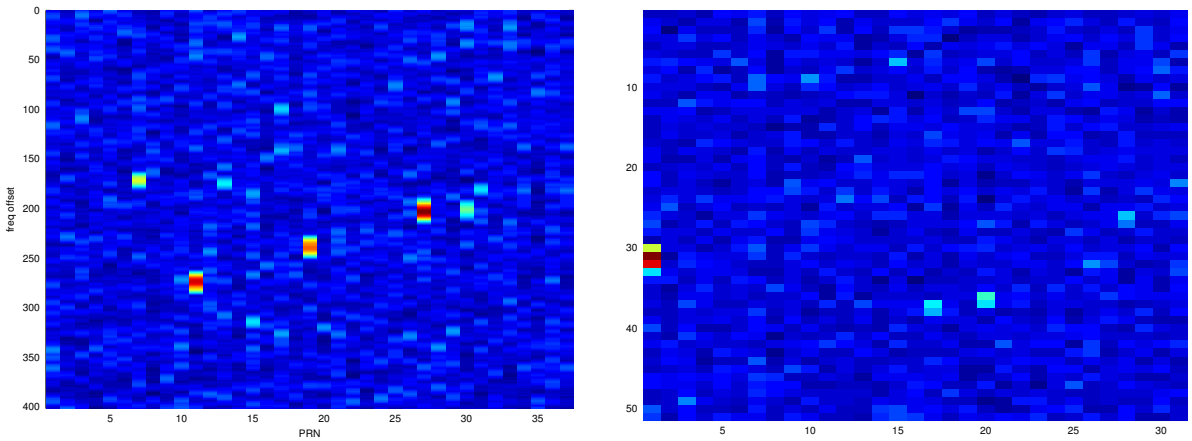


Figure 7: Left: many satellites are visible, including PRN7, PRN11, PRN19, PRN27, and PRN30. Right: only PRN1 provides a powerful signal, while PRN17 and PRN20 are barely visible.

4 Signal acquisition

Let us begin at the beginning: we record a radiofrequency signal in the GPS frequency band, without knowing which satellite is visible nor the position of each satellite in space. We must brute-force search for all possible combinations allowing for the identification of the visible satellites, and the Doppler shift due to their position in the sky.

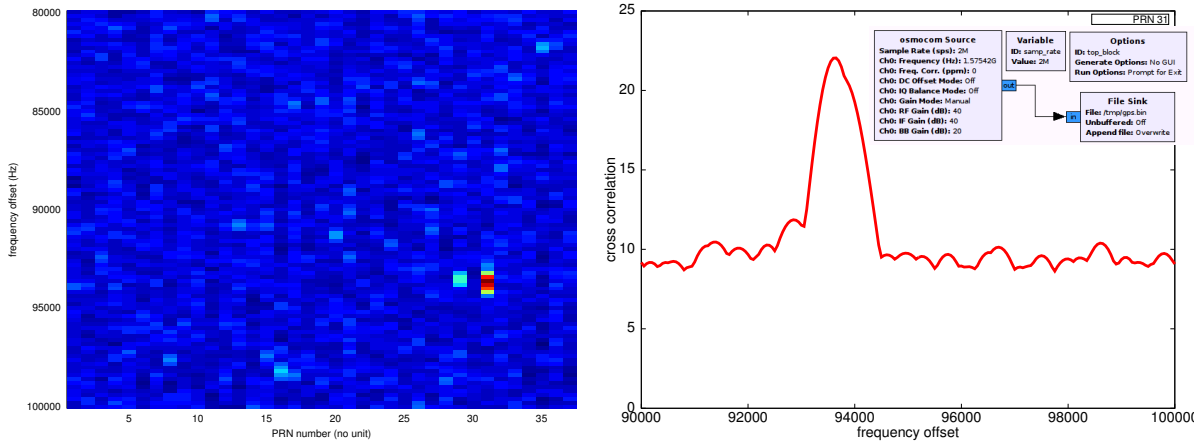


Figure 8: Left: map of visible satellites at a given time (abscissa: PRN, ordinate: frequency offset). Right: evolution of the cross-correlation maximum as a function of the frequency shift for satellite number 31, the one best visible on this record.

Experimentally, we have observed that sweeping the frequency range must be done using 100 Hz steps, otherwise the cross-correlation peak might be hidden in noise since the local oscillator offset to the carrier frequency induces excessive phase drift during the coherent accumulation. For a fixed receiver, and a Doppler shift of at most ±5 kHz, we must hence sweep 100 frequency samples and 31 possible PRN codes. However, we must also account for the huge offset of the local oscillator of the DVB-T receivers: we have observed offsets up to ±60 ppm to the nominal oscillator frequency, a poor performance but hardly surprising for consumer electronics. Such an offset yields a frequency range to be analyzed of ±100 kHz, hence multiplying by 20 the search time during the receiver calibration step. This computation is however only needed once for every DVB-T receiver, and later analysis will only require offsetting the search range by the proper calibrated value (Fig. 8).

We illustrate the identification of satellites over a long measurement sequence by acquiring automatically, every 5 minutes, records lasting one second of radiofrequency signals around the GPS carrier frequency, followed by a search of visible satellites. Automating the acquisition is achieved by replacing the recording step from `gnuradio-companion` with the use of `rtl-sdr` whose arguments are the local oscillator frequency (*i.e.* by which frequency is the signal received on the antenna shifted), the sampling rate and the duration of the record. Practically, we use

```
while true; do name=‘date +%s‘;echo $name; rtl_sdr -f 1575420000 -s 2000000 -g 42 -n 200000 ${name}.bin ;sleep 300;done
```

The file resulting from each acquisition includes data encoded on signed 8 bit values, with interleaved I and Q coefficients. Such a dataset is loaded in GNU/Octave with

```
f=fopen('1426853000.bin');x=fread(f,inf,'uchar');zz=x(1:2:end)-127+i*(x(2:2:end)-127);
```

and the next processing steps are similar to those described earlier. A movie exhibiting the time-evolution, over a total duration of 54 hours, of visible satellites shows the satellite motion since their orbit is below geostationary altitude, and successive satellites being visible over a given point on the surface of the Earth. The dataset size is significant: each 1 s record at a sampling rate of 2 MS/s is 4 MB, so the total 54 h dataset is about 2 GB. The movie resulting from these processing steps is available at `http://jmfriedt.free.fr/gps_movie.avi`.

A detailed analysis example of the files recorded this way is described. A first step is to load the data from the files, remove the average value, define the frequency range (depending on the receiver – here the receiver offset is about -95 kHz to which the ±10 kHz Doppler shift is added and we additionally

account for possible thermal drift) as well as the time axis whose time-step is given by the inverse of he sampling rate `fs`

```
d=dir('./1*.bin');
for k=1:length(d)
  f=fopen(d(k).name)
  x=fread(f,inf,'uchar'); x=x(1:2:end)-127+i*(x(2:2:end)-127); fs=2.0; % MHz
  freq0=[-10.5e4:500:-8.5e4];
  x=x(1:2e5);
  time=[0:1/fs/1e6:length(x)/fs/1e6]';time=time(1:end-1);
```

Having loaded each file, we generate a local copy of each PRN code number `m` in `a`, shift the signal by a frequency `freq` by multiplying the acquired signal with a periodic signal whose phase is `freq*time`, and finally perform the cross-correlation in order to search for the synthesized pattern in the received signal

```
  for m=[1:31]
   a=cacode(m,fs/1.023); a=a-mean(a);
   l=1;
   m
   for freq=freq0 % run through possible frequency offsets
    mysine=exp(j*2*pi*(-freq)*time);
    xx=x.*mysine; % frequency shift the signal
    [u(l,m),v(l,m)]=max(abs(xcorr(a,xx))); % check for cross correlation max.
    l=l+1;
   end
  end
```

Finally, for each file a chart exhibiting the magnitude of the cross-correlation as a function of the code number (abscissa) and the frequency offset (Doppler + thermal drift + bias) in the ordinate axis is plotted:

```
  imagesc([1:31],freq0/1e3,abs(u))
  xlabel('PRN');ylabel('frequency␣(kHz)')
  title(d(k).name)
end
```

Hence, the analysis is summarized as an initial exhaustive search of all possible satellite codes for all possible Doppler shifts: this sums up the acquisition step. Then, knowing which satellite is visible, we will process the phase of the cross-correlation to extract the quantity we are interested in, namely the carrier frequency and the navigation message. Since the frequency keeps on drifting at a rate of about 10 kHz/12 h or [7] 0.23 Hz/s, we must constantly fine tune the carrier frequency for the decoding of the phase of the navigation message to be usable: this feedback loop is the core topic of the *signal tracking* step.

5 Signal tracking

Having roughly (we will see later what that "roughly" means identifying the carrier frequency to within 50 Hz for the feedback loop to lock) identified the carrier frequency and the satellite identifier, we replace the brute-force search (f, PRN) by a feedback loop on the received carrier frequency. Indeed, we might consider that the rough carrier frequency identification is enough and that we can now extract the phase information to search for π rotations and hence find the message (Fig. 9). This is not possible: the phase varies too quickly to allow for information decoding, since the phase rotates as $\Delta f \cdot t$ with Δf the leftover error between the received carrier frequency and the local oscillator frequency. This aspect is demonstrated on Fig. 9 (left), in which we attempted to manually fine tune the local oscillator frequency to cancel the phase rotation either at the beginning or the end of an acquisition lasting a few seconds. The phase shift between the beginning and the end of this dataset is enough to prevent the extraction of the navigation message which is sent at a rate of 50 bps or 20 samples obtained at the output of the

[7] the assumption of a Doppler frequency shift of ±5 kHz during an orbit lasting 12 h does not account for the local oscillator thermal drift.

correlation of the PRN pattern with the received signal. Implementing a feedback loop tracking the local oscillator on the received signal carrier frequency is mandatory for decoding the useful GPS information, or make the best of the local copy of the carrier frequency.

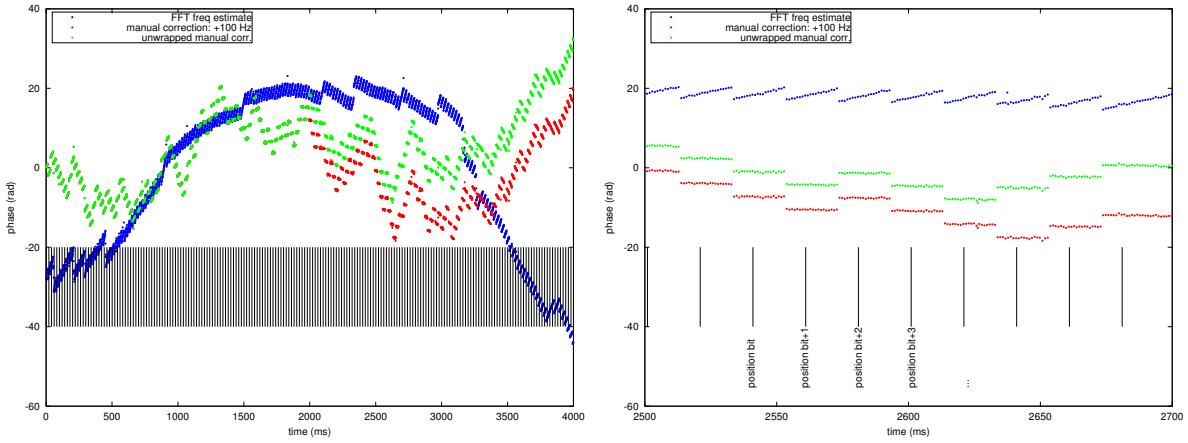


Figure 9: Cross-correlation phase – computed at each cross-correlation magnitude maximum found every millisecond – illustrating the slow offset between the properties of the received signal carrier frequency and the local oscillator, preventing any automated extraction of the navigation message unless a feedback loop is implemented.

The last implementation subtlety lies in the fact that the phase of the carrier has two roles: it holds the transmitted information (jump of π to mark the transition from one bit state to another), and should be used as discriminator if the frequency of the local oscillator deviates from the frequency of the received carrier.

The solution to this dual functionality is to apply a phase estimator which is insensitive to π rotations (instead of 2π as is usually done in trigonometric functions). It happens that by using the $\arctan(Q/I)$ function, we have lost, by using the ratio of Q over I, the information on the individual signs of I and Q. Because of this, the result of arctan always lies in the right part of the trigonometric circle, or in other words is insensitive to 180 degrees phase rotations. This is exactly the property we were looking for (Fig. 10).

Hence, we shall exploit two phase estimators when processing the I and Q coefficients: $\arctan(Q/I)$ to track the carrier frequency, since insensitive to 180^o jumps, and $\text{atan2}(Q, I)$ (as found in Matlab or GNU/Octave) which covers the whole trigonometric circle. The latter function will be used to extract the bits of the navigation message and hence validate our processing steps. Other discriminators insensitive to 180^o phase rotation have been described [13].

Implementing the digital phase-locked loop requires identifying a feedback control which compensates, by varying f the frequency of the digital oscillator $\exp(2\pi jft)$ by which the acquired signal is multiplied, the phase at the cross-correlation between recorded signal and local code copy magnitude maximum (maximums which repeat every millisecond). Furthermore, since the code generator is not synchronized on the satellite generator – if only due to the Doppler shift which also affects the code rate, even if at a lower scale than for the carrier (both carrier and code frequencies exhibit a ratio of 1500 and so does the Doppler shift affecting them) – we must periodically track the code position on the information provided by the carrier. Both feedback loop operate simultaneously and hence exhibit design challenges illustrated in Fig. 14 (left) during failures until a successful result (Fig 14, middle and right) was achieved.

The first feedback control aims at tracking the virtual Voltage Controlled Oscillator (*VCO*) frequency output in order to keep the phase of the received carrier constant. The second feedback control, which will not be detailed here, keeps the position (phase) of the local code copy constant with respect to the received signal: this second law is not described in detail since its effect appears secondary on short (a few second long) samples we are interested in here, and only becomes mandatory on long sequence processing.

The most naive feedback control – known as the proportional control – states that the output frequency of the VCO is proportional to the difference between the carrier frequency and the local oscillator

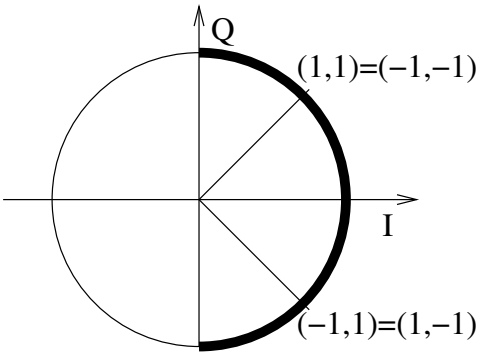


Figure 10: Graphical demonstration of the insensitivity to 180^o phase rotation of the arctan(Q/I) function. Whether I and Q are both positive or negative will yield and angle in the same quadrant. On the opposite, the arctan2(Q,I) uses the individual signs of I and Q to provide and angle estimate over the whole trigonometric circle. The former function is used for carrier frequency tracking, and the latter for extracting the BPSK encoded message.

frequency. A frequency error induces a frequency shift, two quantities with the same unit and the constant between them is unit-less. Such a basic approach fails in practice due to the poor signal to noise ratio of our signals.

In order to "smooth" the observed frequency differences between carrier and local oscillator, the classical approach is to apply a sliding average, also known as an integration, which naturally brings us to the proportional-integral control. The integrator accumulates successive error observations, slowly bringing the frequency of the local oscillator to equate the received signal carrier frequency. When this condition is met, the error becomes null and from then on, the proportional term becomes dominant to keep the VCO frequency equal to the carrier frequency. Notice that for a Doppler shift of ± 4 kHz over 12 h, the carrier frequency variation is 0.2 Hz/s, or a full phase rotation of 2π every 5 seconds, preventing any BPSK demodulation when this feedback control fails.

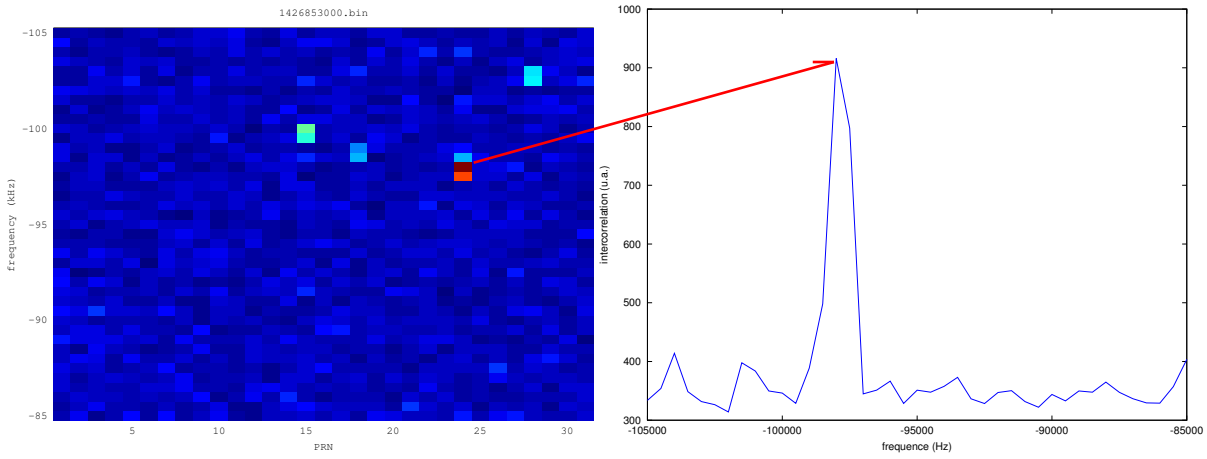


Figure 11: Left: exhaustive search of all possible satellite codes to identify which pattern is usable. Right: evolution of the cross-correlation maximum as a function of local oscillator frequency – an offset of about 97850 Hz maximizes the cross correlation and is used as starting point for the phase tracking.

Practically, the VCO command (which defines its output frequency) u_k results from a linear combination of past errors ε_{k-i} between the carrier signal frequency and the frequency of the VCO, and past values of the command u_{k-i}. Fig. 11 displays the result of the preliminary analysis (acquisition) of a dataset in order to identify which satellite is visible and with which frequency offset. We investigate further the influence of the initial frequency condition on the tracking on Fig. 12, following the rough

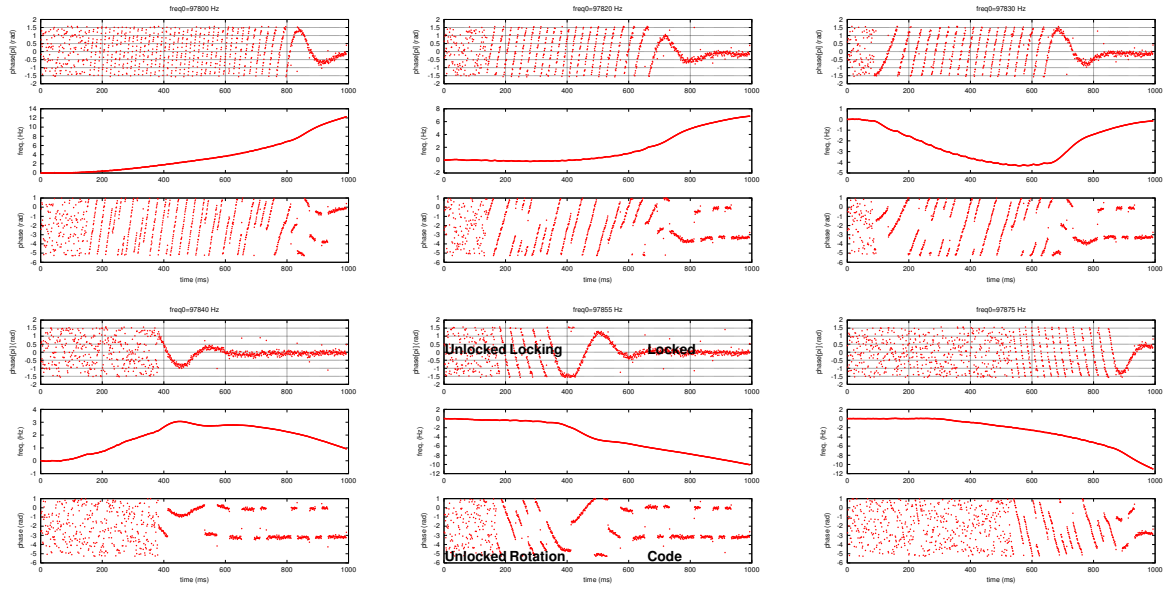


Figure 12: Result when applying the same feedback loop control on various initial conditions of the frequency offset of the local oscillator to the received carrier. For each chart, top is the phase modulo π useful to check the frequency control (error signal), middle is the frequency correction brought to the local oscillator, and bottom is the phase of the signal, in which the navigation message bit values are recovered once the feedback loop control is locked. Initial frequency conditions are respectively 97800, 97820, 97830, 97840, 97855 et 97875 Hz.

yet exhaustive search during the acquisition analysis. Considering how sensitive initial conditions are, identifying the initial frequency with a 50 Hz resolution seems suitable during the acquisition phase. We aim at locking, in this phase locked loop, the cross-correlation phase on the null setpoint so the error signal ε_k is directly equal to the cross-correlation phase φ_k and its past values φ_{k-i}.

A classical approach in control theory relies on determining the operating point by slowly varying the command u_k which rises or decreases depending on the sign and value of ε_k. We select thus u_k as the integral of ε_k, so that $u_k = \alpha \sum_{i=0}^{k} \varepsilon_i$. Writing explicitly the first terms, $u_0 = \alpha\varepsilon_0$, $u_1 = \alpha\varepsilon_0 + \alpha\varepsilon_1 = u_0 + \alpha\varepsilon_1$, $u_2 = \alpha\varepsilon_0 + \alpha\varepsilon_1 + \alpha\varepsilon_2 = u_1 + \alpha\varepsilon_2$, ... the recurrence relation $u_k = u_{k-1} + \alpha\varepsilon_k$ becomes obvious. The value of α remains to be determined: it is usually first set to a small value (hence the control loop slowly varies, with the risk of never allowing for locking the feedback loop before reaching the end of the recorded dataset in this example). Then, α is increased until an acceptable settling time is met, in this case a fast locking of the phase locked loop (Fig. 13). Initial trial and error leads to determine an appropriate value for α, in our $\alpha = 0.01$. We however notice on Fig. 13 that the average value of the phase error remains around -0.29. Such a result is not intrinsically problematic but another type of control loop aims at removing this constant error offset and hence reduce chance of "unlocking" when strong noise is present: the double-integrator, in which u_k is the integral of the integral of ε_k.

We then write

$$\begin{aligned} v_k &= v_{k-1} + \beta\varepsilon_k \\ u_k &= u_{k-1} + v_k + \alpha\varepsilon_k \end{aligned}$$

from which we conclude that:

$$u_k = 2u_{k-1} - u_{k-2} + (\alpha + \beta)\varepsilon_k - \alpha\varepsilon_{k-1}$$

The system is then stable and with a zero-error by selecting α small as before, and $0 < \beta \ll \alpha$.

Hence, the control loop is written as $u_k = 2u_{k-1} + u_k + (\alpha + \beta)\varepsilon_k - \alpha\varepsilon_{k-1}$ in which we must identify values for the α and β coefficients. We have selected $\alpha = 0.0053$ and $\beta = 0.05\alpha$. α and β define the properties of the control loop: too large an α yields too fast a feedback loop which might overshoot the

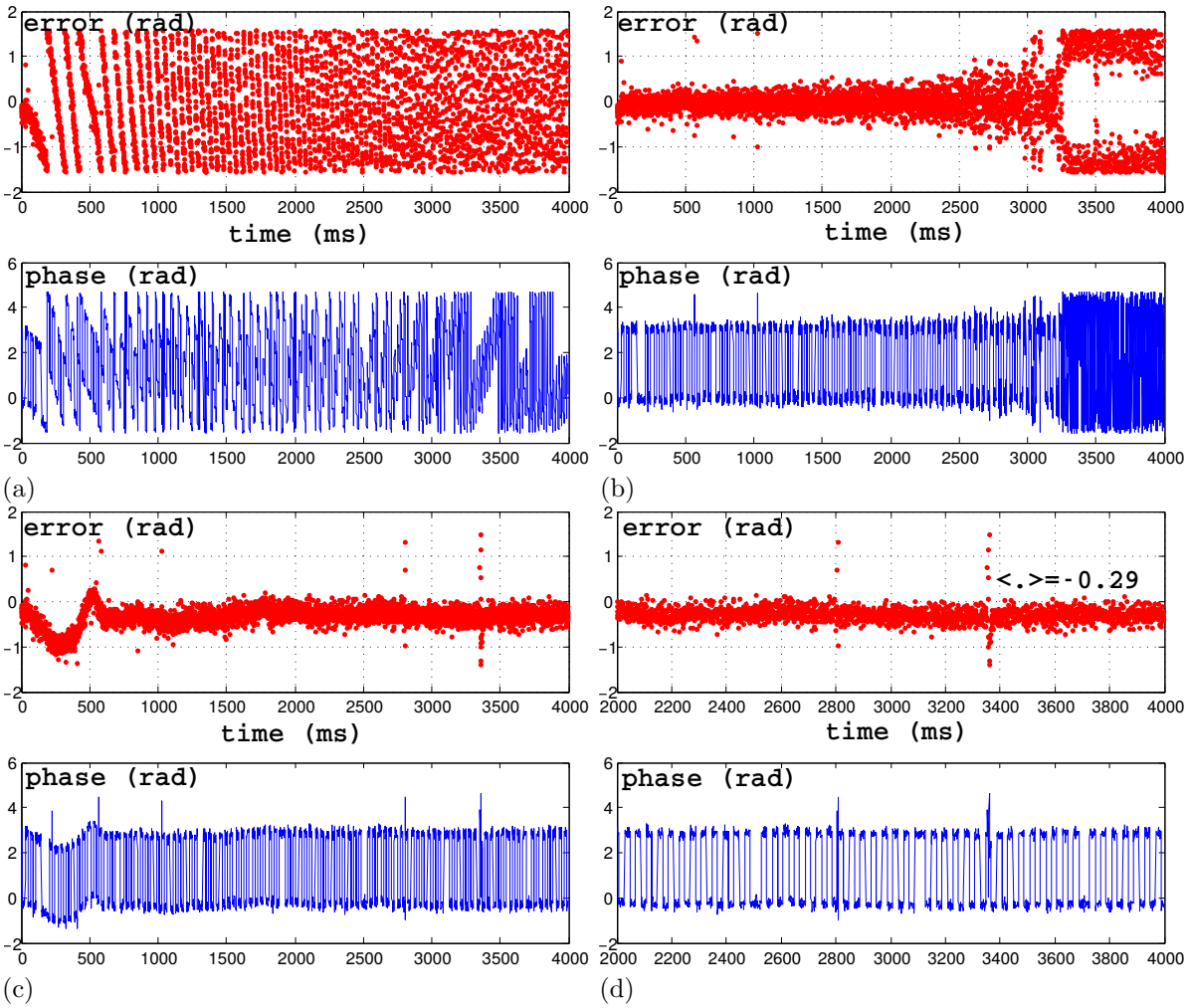


Figure 13: Influence of the value of α on the convergence of the feedback control, with the phase modulo π in red (top) on each chart indicative of the control of the VCO, and (bottom) the unwrapped phase including the navigation data. (a) $\alpha = 0.001$, the feedback is not fast enough to lock. (b) $\alpha = 0.1$, the feedback loop locks but becomes unstable (α gain too big). (c) For $0.005 < \alpha < 0.05$ the feedback loop locks, here $\alpha = 0.01$. (d) Zoom on the error demonstrates a residual average phase equal to -0.29 rad.

setpoint and hence yield to locking failure, while too small an α will slow down convergence. β defines the relative weight of past and current values of the cross-correlation phase. In order to assess the influence of each one of these coefficients, Fig. 13 provides and example in which the α coefficient is varied below and above the optimum value of 0.0053, from 0.0070 to 0.0035.

Proper operation of this algorithm is demonstrated by displaying the navigation message, extracted from the phase computed by using `atan2` (hence sensitive to π phase rotations), as shown on Figs 12, 13 and 14.

Practically, we load the dataset from the recorded file, subtract the average value from the I and Q coefficients, and alternate the bytes to generate the `zz` signal to be processed

```
f=fopen('1426853000.bin');
x=fread(f,inf,'uchar');
zz=x(1:2:end)-127+i*(x(2:2:end)-127); fs=2.0; % MHz
```

This procedure for loading unsigned bytes is valid if the acquisition was performed using `rtl-sdr`. If the file was generated through a `File Sink` from `gnuradio-companion`, then the data are stored as floating point values (and not 8-bit integers – hence a file size 4 times larger), the data are loaded using

`read_complex_binary` provided in the `gr-utils` directory from the GNURadio source code.

```
zz=read_complex_binary('1426853000_grc.bin');
```

We will focus on satellite number 31 (as identified during the previous acquisition phase) and considering an initial frequency offset between received carrier and local oscillator of about 93705 (position of the cross-correlation maximum found during the previous step)

```
msatellite=31; % satellite number definition
freq0=-(93705);

N=1; % starting point for the measurement
Nb_points=30790; % duration of the analysis in ms
x=zz(N+1:N+Nb_points*1000);
```

The time axis is defined with steps equal to the inverse of the sampling rate – here 2 MHz – and various parameters (`alpha`, `beta`) of the control loop are defined. The control loop parameters are dependent on the sampling rate of the acquired signal: following the cross-correlation, we obtain one measurement every millisecond, so the sampling period is 1 ms. The control loop is defined as a recursive relationship between a set of weighting coefficients of past measurements (phase – `e_k`) and weighting coefficients of past control values (VCO output frequency – `u_k`).

```
time=[0:1/2e6:(length(x)-1)/2e6]';

% % "minimum" control = 1 integrator -> static error
% % u_k=u_k-1 + alpha*e_k
% alpha=0.05;
% coef_uk_1 = 1;
% coef_uk_2 = 0;
% coef_ek = alpha;
% coef_ek_1 = 0;

% double integrator -> no static error left
% u_k=2u_k-1 -u_k-2+ (alpha+beta)*e_k -alpha*e_k-1
alpha=0.01;beta=0.001;
coef_uk_1 = 2;
coef_uk_2 = -1;
coef_ek = alpha+beta;
coef_ek_1 = -alpha;
```

After initializing a few additional variables – most significantly GNU/Octave and Matlab love to know from the beginning the size of the matrices they handle and not to dynamically allocate memory – the PRN code of the selected satellite (`msatellite`) is generated, and pas values of the control and measurements are set.

```
% initialisation des variables pour la vitesse de calcul
fr=NaN(Nb_points/2,1);
yp=NaN(Nb_points/2,1);
yyp=NaN(Nb_points/2,1);

% quelques initialisations
freq=0;
freqm1=freq;
freqm2=freqm1;
ym1=0;
l=1;

ap=cacode(msatellite,2/1.023); ap=ap-mean(ap);
al=[ap(end) ap(1:end-1)];
ae=[ap(2:end) ap(1)];
```

Two control loops are needed when analyzing the dataset: carrier frequency and pseudo-random code position in the acquired dataset. The former is taken care of by analyzing the phase, modulo π, of the

received signal, while the latter will result from the comparison of local copies of the pseudo-random pattern delayed (`l` as in *late*), in advance (`e` as in *early*) or current (`p` as in prompt) during the cross-correlation. The pattern is slowly shifted to track the phase induced by the slight shift between our clock used to generate the sampling rate and the satellite clock used to generate the identifier code.

```
% main loop
for kk=1:2000:length(x)-2001
  mysine=exp(1j*2*pi*mod((freq0+freq)*time(kk:kk+2000),1)); % frequency offset
  xx=x(kk:kk+2000).*mysine;

  zl=xcorr(al,xx); % l=late ; p=present ; e=early
  zp=xcorr(ap,xx);
  ze=xcorr(ae,xx);

  [aal,bbl]=max(abs(zl));
  [aap,bbp]=max(abs(zp));
  [aae,bbe]=max(abs(ze));

  % coherent discrimanator: controls the position of the PRN code
  d=(abs(ze(bbe)^2)-abs(zl(bbl))^2)/(abs(ze(bbe))^2+abs(zl(bbl))^2);
  % shift PRN code by 1 bit depending on the value of the coherent discriminator
  if d<0
    ap=[ap(end) ap(1:end-1)];
    al=[ap(end) ap(1:end-1)];
    ae=[ap(2:end) ap(1)];
  else
    ap=[ap(2:end) ap(1)];
    al=[ap(end) ap(1:end-1)];
    ae=[ap(2:end) ap(1)];
  end
```

Having aligned the code position, we are now ready to start the frequency tracking loop following all these initializations, feedback loop which aims at keeping the local oscillator frequency equal to the received carrier frequency (identified as the phase, modulo π, of the received signal), and on the other hand keep the local replica of the code aligned with the signal received from the satellite. Once these two tracking loops are locked, a phase estimate (this time modulo 2π and not π) displays the status of the navigation message – 0 and 1 represented by angle of 0 and π, in the variable `yyp`:

```
  [valeur,index_max]=max(abs(zp));
  yp(l)=atan(imag(zp(index_max))./real(zp(index_max))); % carrier=phase [pi]
  yyp(l)=atan2(imag(zp(index_max)),real(zp(index_max))); % bits =phase [2pi]

  % application of the recurrent equation of the controller
  freq=coef_uk_1*freqm1+coef_uk_2*freqm2+coef_ek*yp(l)+coef_ek_1*ym1;

  fr(l)=freq;
  freqm2=freqm1;
  freqm1=freq;
  ym1=yp(l);

  l=l+1;
end
```

Finally, the result of this analysis is displayed as the phase of the signal after mixing with the controlled VCO, the frequency of the VCO, and the phase of the signal including the navigation information.

```
subplot(311)
plot((yp),'r.');
ylabel('phase[pi] (rad)');
eval(['title(''freq0=',num2str(freq0),' Hz ;\alpha=',num2str(alpha),''');']);
grid
subplot(312);plot(fr,'r.');ylabel('freq. (Hz)');
```

```
kyyp=find(yyp>1);yyp(kyyp)=yyp(kyyp)-2*pi;
subplot(313);plot((yyp),'r.');ylabel('phase_LJ(rad)');xlabel('time_LJ(ms)')
save yyp yyp % save the binary code extracted from the demodulation
% eval(['print -depsc 1426853000_',num2str(freq0),'_',num2str(round(alpha*1000)),'.eps']);
```

Despite being a fundamentalist of using opensource software, it is clear the the proprietary tool Matlab yields results much faster than GNU/Octave. Indeed, processing 1.4 million samples (700 ms duration, since each millisecond includes 2000 points when sampling the radiofrequency signal at a rate of 2 MS/s) lasts 61 s with Matlab (R2010) against 371 s with GNU/Octave (3.8.2).

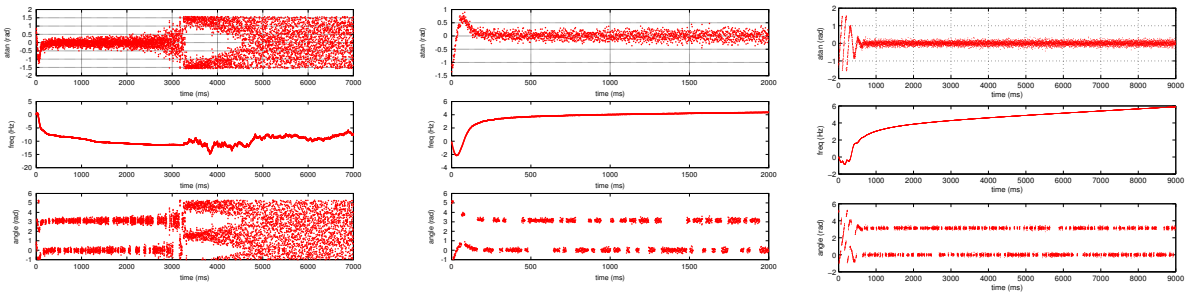


Figure 14: On the three charts, from top to bottom: atan(Q/I) for controlling the carrier phase ; the frequency of the local oscillator ; arctan2(Q, I) for extracting the phase modulation, illustrating the divergence of the feedback loop after processing a few seconds worth of data, even though decoding works fine in the beginning of the processed dataset. Left: signal recorded with a DVB-T receiver clocked by its internal quartz oscillator. Middle: the receiver is clocked by the output of a Rohde & Schwartz SMA 100 synthesizer in order to assess the influence of the quartz oscillator instability on the ability to extract the navigation message. Right: the longest message we were able to decode, lasting a total duration of 9 seconds.

We conclude this study of decoding the GPS navigation message by looking for the header of each message, the only constant part that repeats every 6 seconds (beginning of the transmission of a new word). The synchronization word, named TLM, is defined as the 10001011 byte [2, p.112], which we search for in the bit sequence resulting from the processing steps developed earlier. Since the pattern transmitted by the satellites (CDMA) is repeated every millisecond and the digital transmission is clocked at 50 bits/second, we improve the identification capability of each bit by averaging over 20 successive values after finding the first sharp transition from 0 to 1. Finally, the TLM pattern is searched for in the bit sequence by cross-correlation [2, p.119] : proper decoding of the sequence can only be validated if this cross-correlation maximum repeats every 6 seconds (in addition to the random occurrences of the pattern in the message). By decoding 15 seconds worth of telemetry, we indeed recover two cross-correlation maximums located at dates 4.4 and 10.4 s, demonstrating our processing steps are sound. Obviously, such a short code as TLM only provides a poorly resolved correlation peak with respect to the background noise: the cross-correlation maximum is 2 (`tlm=[1 0 0 0 1 0 1 1];sum((tlm-mean(tlm)).^2)`) and all cross-correlation values lie on 8 possible values, which are reduced to 4 once the absolute value of the cross-correlation is considered (the cross-correlation is equal to -2 if we have inverted the 0 and 1 states of the navigation bit values, *i.e.* we have inverted the 0 and π phase values of the decoded signal).

The GNU/Octave program yielding the chart shown on Fig. 15 is detailed below

```
load yyp % reload file with the phase output [2pi] = BPSK code
l=1
for k=987:20:length(yyp)-20 % 987 = first usable phase transition
  sortie(l)=sum(yyp(k:k+19)); % sliding average over 20 samples, synchronized on 1st transition
  l=l+1;
end
temps=[0:1/50:length(sortie)/50];temps=temps(1:end-1); % time axis
k=find(sortie>-30);sortie(k)=0; % threshold to 2 binary values
k=find(sortie<-30);sortie(k)=1;
symbole=[1 0 0 0 1 0 1 1];symbole=symbole-mean(symbole);
sortie=sortie-mean(sortie); % always work with data with mean value =0 when using
```

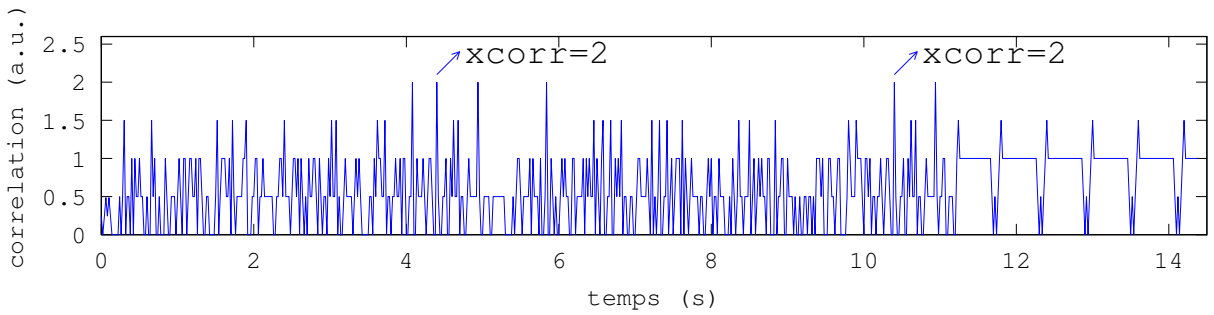


Figure 15: Cross-correlation of the TLM synchronization byte with the bit sequence resulting from GPS signal processing: the repetition every 6 seconds of the cross-correlation maximum proves the processing is sound. Notice that this 15 second long GPS record requires a storage space of 60 MB (or 240 MB when `gnuradio-companion` stores samples as floating point numbers instead of integers) !

```
12 r=(abs(xcorr(symbole,sortie))); % the cross-correlation to find a pattern
13 r=r(1:length(temps));
14 plot(temps,r); % graphical chart
15 axis([0 14.5 0 2.6])
16 text(10.6,2.3,'xcorr=2');drawArrow(10.4,2.1,10.7,2.4,0.1,0.1)
17 text(4.6,2.3,'xcorr=2') ;drawArrow(4.4,2.1,4.7,2.4,0.1,0.1)
18 xlabel('temps␣(s)');ylabel('correlation␣(a.u.)')
```

while the content of the `sortie` variable is analyzed in detail (here 3 successive sentences, the latter two being annotated to help the reader) with

```
10101010101010100101010101010101010101010101011010101010101010101010101010101001
01010101010101010101010101011010101010101010101010101010101010010101010101010101
0101010110011100
  TLM        reserved     parity      HOW-count        ID     parity  DATA
10001011 0000110001010100 000100 01011101111000110 0 1 101 01 010100 010011100
10000001100001111011110011100111011101111100111111100000010110001001111111111110
11010100001000011010111000011111101001101111100100011011110011101010000011101O
11100000100100101011001111001110001011010011111101111000000000000001000000000
  TLM        reserved     parity      HOW-count        ID     parity  DATA
10001011 0000110001010100 000100 01011101111000111 0 1 001 10 101000 110001010
10100000000000011111110110001010100000000110110100110011010111010100100000001O
01101011001001100011110010011011101111010111100000011011111111100111001011011
11110110111000000000000
```

The beginning of the sequence is a leftover of the previous sentence. The synchronization byte (10001011) follows, then the 16 bit-long reserved sequence and the 6 parity bits of the first 30 bit word, and the HOW. Only its ID – which identifies which kind of sentence is being transmitted (between 1 and 5) – is of interest to us (notice though the increment by 1 of the HOW-count counter). We indeed observe an identifier equal to 5 (101 : end of the previous sentence) followed, in the next sentence, by 1 (beginning of a new sentence). The last two bits of HOW (second word, included in the last two bits f the HOW (second word, included in the end of the 6-bit parity byte) are indeed equal to 0: all this description is consistent with the navigation message description found at `http://www.losangeles.af.mil/shared/media/document/AFD-070803-059.pdf` (pp.109 and followings). For the sentence of identifier 1, the first 10 bits indicate the transmission date modulo 1024. We find 1100010101 which describes week number 789 and an acquisition between October 5th and 11th in 2014 (cf `http://adn.agi.com/GNSSWeb/`), consistent with the experiments we performed. Follow two bits concerning the transmission status – 01 describes the code being transmitted on the second channel on the L2 carrier, then the accuracy of the positioning when using data from this satellite: the best possible resolution is achieved, between 0 and 2.4 m, since all systems are operational (all bits equal to 0). This analysis hence demonstrates reliable decoding of the navigation message following the processing steps described in this document.

6 Conclusion

Despite long term interest in decoding GPS signals, it is only recently that all the elements needed to achieve this task on a shoesting budget became available including a low cost radiofrequency receiver operating at 1.5 GHz, with high enough bandwidth for decoding a signal transmitted at a rate of a Mb/s, and the computational processing power needed to implement the decoding software. This cunjuncture is the opportuniy for any curious mind to reproduce these experiments. The initial demonstration of the feasibility of such a project was demonsrated during P. Boven's presentation at OHM2013 entitled "Hacking GPS", in which we were told that squaring the I/Q coefficients to get rid of the modulation which is the cause of spectrum spreading, would for the coherent accumulation of carrier signals received from the satellites while lowering the thermal noise. The practical implementation of this idea is not as trivial as this talk would have one think, most significantly because we must know at what frequency offset to look for the signal of interest, next to all the spurious and parasitic peaks of the Fourier transform. After quite some investigation, we inded up finding a signal demonstrating the radiofrequency emission of GPS satellites using this method (Fig. 16). This method appears though, despite seemingly simple, to provide poor results difficult to analyze. It was nevertheless the beginning of this investigation, and deserved being mentioned in the conclusion of a study demonstrating the thorough understanding of the signal nature and of their modulation.

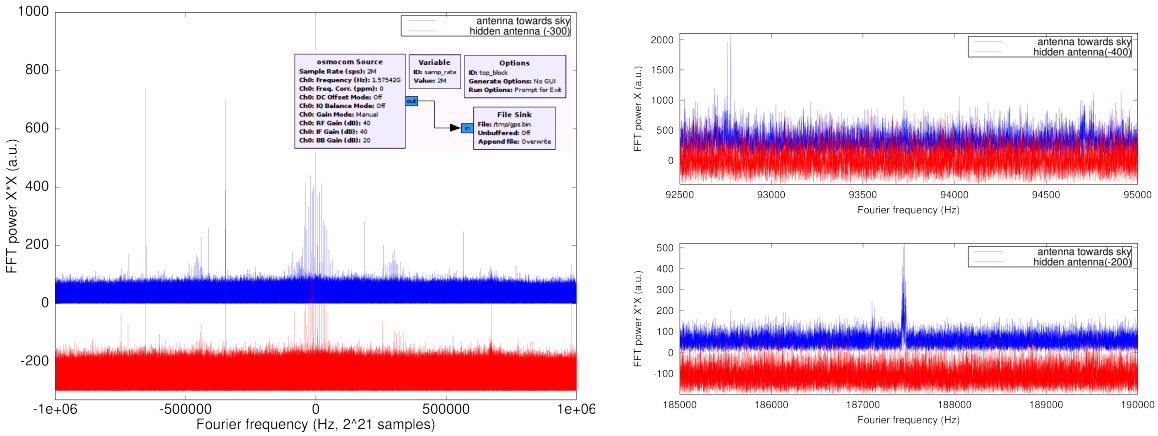


Figure 16: Left: wideband analysis if the squared $I + jQ$ coefficients, for an antenna facing the sky (blue) or hidden from the sky by a metallic plate (red). Many parasitic modes make the analysis if this signal complex. Right: zoom on the region including the double of the frequency offset between the local oscillator and the carrier from the signal coming from the GPS: top the Fourier transform of the $I + jQ$ signal does not exhibit any relevant feature due to the spectrum spreading introduced by the pseudo-random modulation, but squaring this signal (bottom) allows for the coherent accumulation of the radiofrequency signal coming from the satellite and generate a significant feature when the antenna is facing the sky (blue), which disappears when the antenna is hidden behind a metallic plate (red). These acquisitions were realized with an E4000 receiver, more noisy than the R820T frontend.

What next ? Many issues remain to be solved. The first one is that we have been able to display and analyze bits from the navigation message, but we have not gone all the way through actually using the ephemeris of the satellites to finally locate the receiver in space. This step appears mandatory even for the less ambitious purpose of reproducing on the ground a frequency source as stable as a combination of the embedded atomic clocks, since it seems the only way of compensating for the Doppler shift which affects all observed carriers. Additionally, using the shape of the correlation signals to extract from a comparison of the direct and reflected paths the distance of the receiver to a reflecting target (passive RADAR application) appears complex for ground based measurements considering the restricted bandwidth of our measurements. Indeed, with a code repeating at a rate of 1.023 MHz, one time interval of the correlator lasts 977 ns during which an electromagnetic signal propagates 294 m. The reflector must be located at several times this distance for the secondary cross-correlation peak – due to the reflected signal and not the direct signal – to be clearly separated and easy to detect. This explains

why most applications exploiting GPS reflected signals for a distance measurement are performed using airborne receivers, and additionally over the sea in order to have a wide reflector and hence improve the signal to noise ratio. Such fields of applications are beyond the means accessible to the amateur. Finally we have not been successful in decoding signal generated by geostationary satellites to help navigation – EGNOS in Europe [8] and WAAS over Americas – even though their code and communication protocol should be compatible with GPS. The cause of this failure is has not been identified.

An archive with some datasets and GNU/Octave scripts is available at `http://jmfriedt.free.fr/efts_archive.tar.gz`

Acknowledgements

This study was completed as part of the lab courses of École d'Été sur le Temps-Fréquence (EFTS) held in Besançon. Financial support for hosting an ERASMUS student – Sarà Martinez Gutierrez – has been provided by labex First-TF (`http://first-tf.com/`). Literature references which are not freely accessible on the web have been fetched on Library Genesis, `gen.lib.ruc.ec`: our research activities would not be possible without such a resource.

J.-M Friedt is assistant professor at Université de Franche Comté in Besançon (France), consultant for a private company manufacturing passive wireless sensors interrogated by electronic systems acting similarly to RADARs. G. Cabodevila is assistant professor at École Nationale Supérieure de Mécanique et Microtechniques in Besançon (France). His course on feedback loops and control theory, which provides a formal framework to the concepts introduced in section 5, is available at `http://jmfriedt.free.fr/Gonzalo_cours1A.pdf` [in French]. Both are hosted for their research activity by the FEMTO-ST institute.

References

[1] J.-M Friedt, *La réception de signaux venus de l'espace par récepteur de télévision numérique terrestre*, OpenSilicium **13** (Dec 2014/Jan-Fev 2015), available at `http://jmfriedt.free.fr/sdr2.pdf` [in French]

[2] K. Borre, D.M. Akos, N. Bertelsen, P. Rinder, & S.H. Jensen, *A software-defined GPS and Galileo receiver – A single frequency approach*, Springer (2007)

[3] E.D. Kaplan, *Understanding GPS: Principles and Applications, 2nd Ed*, Artech House Mobile Communications (2005)

[4] R.L. Beard & J.D. White, *GPS Application to Time Transfer and Dissemination*, GPS Solutions **3** (1), pp.17–25 (1999)

[5] C. Bruyninx, P. Defraigne & J.-M Sleewaegen, *Time and Frequency Transfer using GPS Codes and Carrier Phases: Onsite Experiments*, GPS Solutions **3** (2), pp.1–10 (1999)

[6] Program and slides of the Journée GNSS 2015 organized by CNES, `http://geodesie.ign.fr/journee-gnss-science/programme/` [mostly in French]

[7] an excellent book summarizing the initial developments which lead to defining the time on atomic transitions to replace the astronomical definition, mandatory reading: T. Jones, *Splitting the second – The Story of Atomic Time*, Institute of Physics Publishing (2000)

[8] G. Lachapelle, *GNSS Solutions: Atomic clocks on satellites and mitigating multipath*, Inside GNSS (Sept. 2006), disponible à `http://www.insidegnss.com/auto/Sept06GNSS_Solutions_secure.pdf`

[9] `http://gpsinformation.net/main/gpspower.htm` although this value rather seems the result of solving the inverse problem since the GPS specification defines the minimum power *received* on the ground rather than the power emitted by the satellite. Such an approach is discussed in `www.unoosa.`

[8] the appropriate code generator is available at `http://cu-hw-gps.googlecode.com/svn/matlab/waas_enabled_receiver/cacodegn_waas.m`

org/pdf/icg/2010/ICG5/wgA/05.pdf and indeed yields 27 to 28 W (14 dBW – notice the probable unit error in slide 17). www.nxp.com/documents/other/75016740.pdf discusses an emitted power of 50 W.

[10] J.-M Friedt, *Auto et intercorrélation, recherche de ressemblance dans les signaux : application à l'identification d'images floutées*, GNU/Linux Magazine France **139** (Juin 2011), disponible à http://jmfriedt.free.fr/xcorr.pdf [in French]

[11] Y. Guidon, *Sciences/Théorie de l'information : propriétés et dérivés des CRC et LFSR*, GNU/Linux Magazine **085** (Juillet 2006) [in French]

[12] R. Boudot *et al.*, *A high-performance frequency stability compact CPT clock based on a Cs-Ne microcell*, IEEE Trans. Ultrason. Ferroelectr. Freq. Control. **59** (11), pp.2584–7 (2012) ou http://members.femto-st.fr/sites/femto-st.fr.laurent_larger/files/content/Homepage_ll_fichiers/pdf_files/FRGLS13/mo1700boudot.pdf

[13] C. O'Driscoll, M.G. Petovello & G. Lachapelle, *Choosing the coherent integration time for Kalman filter-based carrier-phase tracking of GNSS signals*, GPS Solutions **15**, pp.345–356 (2011)