Digital Signal Processing Laboratory

Lab report 1: Time-Delay estimation in GNSS

Albert Iepure

Student ID: 03650065

MSCE

30.05.2014

I. Theoretical background

In GNSS (Global Navigation Satellite Systems) terminology, the first step that is done by the receiver is to scan for all the satellites in his field-of-view. The purpose of acquisition is to determine visible satellites and coarse values of carrier frequency and code phase of the satellite signals.

The satellites are differentiated by the 32 different orthogonal PRN sequences, which uniquely determines a satellite. All these codes are stored within the receiver. The code phase is the time alignment of the PRN (Pseudo Random Noise) code in the current block of data. The code phase is needed in order to generate a local PRN code that is perfectly aligned with the incoming code. Only when this happens, the incoming code can be removed from the signal. PRN codes have high correlation only for zero time-shift which means that the two signals must be perfectly aligned to remove the incoming code. The third parameter is the carrier frequency, which in case of down conversion corresponds to the IF, which is known for the L1 carrier frequency of 1575.42 MHz and from the mixers in the down converter. However, the frequency can deviate from the expected value because the velocity component of the satellite in the direction of the receiver causes a Doppler shift determining a higher or lower frequency than the expected one. In the worst case, the frequency can deviate up to±10 kHz. It is important to know the frequency of the signal in order to be able to generate a local carrier signal. This signal is used to remove the incoming carrier from the signal. In most cases it is sufficient to search the frequency range such that the maximum error will be less than or equal to 500 Hz.

There are three acquisition algorithms through which the delay τ can be found, namely: serial, parallel frequency space, parallel code phase. In the following I will briefly present each one of them.

Serial search acquisition

The algorithm is based on the multiplication of locally generated PRN code sequences and locally generated carrier signals. The PRN generator generates a PRN sequence corresponding to a specific satellite. The resulted generated sequence has a certain code phase, from 0 to 1022 chips. The incoming signal is initially multiplied by this locally generated PRN sequence. After multiplication with the PRN sequence, the signal is multiplied by a locally generated carrier signal. Multiplication with the locally generated carrier signal generates the in-phase signal I, and multiplication with a 90◦ phase-shifted version of the locally generated carrier signal generates the quadrature signal Q. The I and Q signals are integrated over 1 ms, corresponding to the length of one C/A code, and finally squared and added. However, the I signal generated at the satellite does not necessarily correspond to the demodulated I, because the phase of the received signal is unknown. The output is a value of the correlation between the incoming signal and the locally generated signal. If a predefined threshold is exceeded, the frequency and code phase parameters are correct, and the parameters can be passed on to the subsequent tracking algorithms.

The serial search algorithm performs two different sweeps: a frequency sweep over all possible carrier frequencies of IF±10 kHz in steps of 500 Hz and a code phase sweep over all 1023 different code phases. All in all, this sums up to a total of:

$$\underbrace{1023}_{\text{code phases}} \underbrace{\left(2\frac{10{,}000}{500}+1\right)}_{\text{frequencies}} = 1023 \cdot 41 = 41{,}943 \text{ combinations.}$$

Obviously, this is a very large number of combinations. This exhausting search routine also tends to be the main weakness of the serial search acquisition, which is very time-inefficient.

Parallel Frequency Space Search Acquisition

Because of the time ineffectiveness of the previous procedure, the main idea to make it faster is to eliminate from the search procedure one of the two parameters or to implement them in parallel.

As the name suggests, this second method of acquisition parallelizes the search for the one parameter. The incoming signal is multiplied by a locally generated PRN sequence, with a code corresponding to a specific satellite and a code phase between 0 and 1022 chips. The resulting signal is transformed into the frequency domain by a Fourier transform. The Fourier transform could be implemented as a Discrete Fourier Transform (DFT) or a Fast Fourier Transform (FFT). The FFT is the fastest of the two, but its disadvantage is that it requires an input sequence with a radix-2 length (power of two length).

In parallel frequency space search acquisition, with a perfectly aligned PRN code, the output of the Fourier transform will show a distinct peak in magnitude. The peak will be located at the frequency index corresponding to the frequency of the continuous-wave signal and thereby the frequency of the carrier wave signal.

The serial search acquisition method steps through all possible code phases and carrier frequencies, the parallel frequency space search acquisition only steps through the 1023 different code phases. This comes at the cost of a frequency domain transformation with each code phase.

Parallel Code Phase Search Acquisition

As seen from the equation on the previous page, the amount of search steps in the code phase dimension is significantly larger than that of the frequency dimension (1023 compared to 41). The previous method parallelized the frequency space search eliminating the necessity of searching through the 41 possible frequencies. If the acquisition could be parallelized in the code phase dimension, only 41 steps should be performed compared to the 1023 in the parallel frequency space search acquisition algorithm. This acquisition technique makes use of the convenience to make a circular cross correlation between the input and the PRN code without shifted code phase.

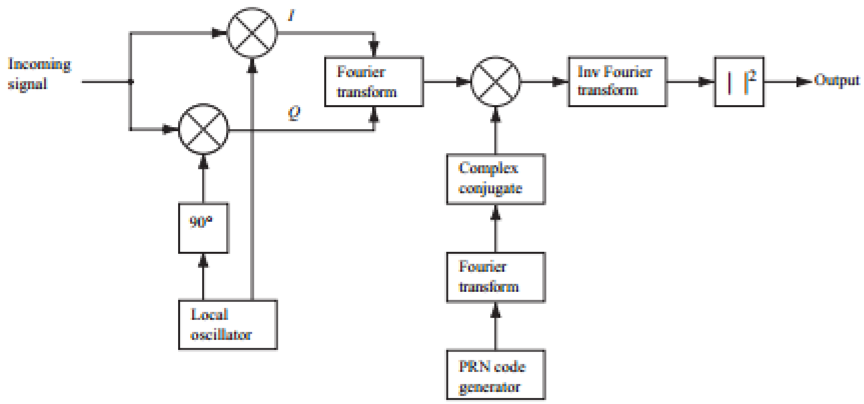


Figure 1.1: Block diagram of the parallel code phase search algorithm

Let x(n) and y(n) be two finite length sequences of length N, periodic, and X(k) and Y(k) the corresponding Fourier transforms. The circular cross correlation sequence between x(n) and y(n) is z(n):

$$z(n) = \frac{1}{N}\sum_{m=0}^{N-1} x(m)y(m+n) = \frac{1}{N}\sum_{m=0}^{N-1} x(-m)y(m-n).$$

Omitting the scaling factor 1/N, the N-point Fourier transform of z(n) is:

$$Z(k) = \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} x(-m)y(m-n)e^{-j2\pi kn/N} = \sum_{m=0}^{N-1} x(m)e^{j2\pi km/N}\sum_{n=0}^{N-1} y(m+n)e^{-j2\pi k(m+n)/N} = X^*(k)Y(k).$$

When the frequency domain representation of the cross correlation is found, the time-domain representation can be found through inverse Fourier transform.

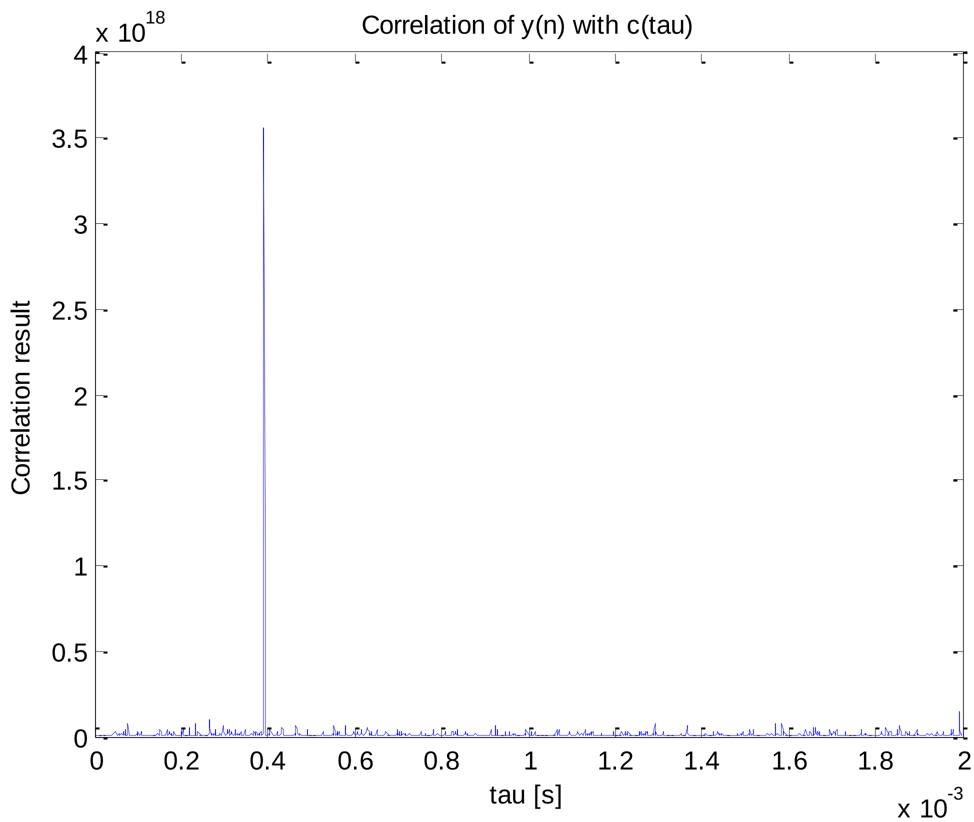
$$z(n)^2 = \mathrm{IFFT}\{X^*(k)Y(k)\}^2$$

The absolute value of the output of the inverse Fourier transform represents the correlation between the input and the PRN code. If a peak is present in the correlation, the index of this peak marks the PRN code phase of the incoming signal.

Compared to the previous acquisition methods, the parallel code phase search acquisition method has cut down the search space to the 41 different carrier frequencies. The Fourier transform of the generated PRN code must only be performed once for each acquisition. For each of the 41 frequencies we perform one Fourier transform and one inverse Fourier transform, so the computational efficiency of the method depends on the implementation of these functions. An idea of implementing the FFT in a more time-efficient manner would be to use the Overlap and Add or Overlap and Save methods.

II. Procedure

By implementing the algorithm described above, the code with τ=0 has to be generated and transformed into the frequency domain, together with the received signal. Afterwards, the Fourier transform of the received signal is multiplied with the complex-conjugate of the Fourier transform of the code, and the result is back-transformed into the time-domain. The result is plotted in the figure below. The point where the peak occurs is actually the delay τ we were looking for, which in this case is 0.400 Tc (0.39101 ms), where Tc is the chip duration, namely the inverse of the sampling frequency. The calculation of τ took 0.038671 seconds, which is very time-efficient because it uses Matlab's in-built functions, which are far quicker than my implementation of overlap-and-add and overap-and-save.



III. Matlab code

```
%% Project 1: Time-Delay Estimation in GNSS %%
% you do not need the code file %

%% Load measurements: received signal
load('proj_1_sample.mat');

%% Correlation
% sampling rate
f=1.023e6;

% Calculation
% Here goes your code

%Parallel Code Phase Search Acquisition
pace=1/f;
tau = 0:pace:size(REC_SIGNAL)*pace-pace;
y_freq_domain = (fft(REC_SIGNAL));
[time, c] =gnss_signal(0);
c_freq_domain= conj(fft(c));
correlation = (ifft(y_freq_domain.*c_freq_domain)).^2;

[maxCorr,maxA] = max(real(correlation));
tau_shift = tau(maxA);
tau_shift_chips = tau_shift/pace;

%% Please print your result

figure
plot(tau,correlation); title('Correlation of y(n) with c(tau)'); xlabel('tau
[s]'); ylabel ('Correlation result');
```

IV. Bibliography

A Software-Defined GPS and Galileo Receiver: A Single-Frequency Approach (Applied and Numerical Harmonic Analysis) by Kai Borre, Dennis M. Akos