Lab 2

Daniele Borio

August 23, 2012

The following materials have been provided with this lab:

- short base-band dataset
- some basic Matlab scripts for tracking GPS L1 C/A signals

The main processing is performed by the myTracking.m script that produces a text file containing the tracking results (correlator outputs, Doppler frequency, code Doppler, etc.). The lab assignment is divided in exercises that require the modification of the main tracking file and the addition of new functionalities.
The first exercise will be developed during the class hours and will serve as a basic example for the solution of the remaining exercises.
A report describing your work must be submitted by Monday 26th August 2010 (at 9:00 a.m., Calgary time). The report must be submitted in electronic form.
Make sure to include any plots that you deem relevant in the report in addition to your observations and answers to the questions below.

Exercise No. 1: Histogram Synchronizer

The myTracking.m script is able to track GPS L1 C/A signal using either 1 or 20 ms coherent integration. In order to enable 20 ms integration, implement a bit synchronizer using the histogram method. The following variables should be part of the bit synchronizer:

- bins: vector of 20 elements containing the number of changes observed in the different code bins;
- currentBin: index pointing to the current bin;
- previousPrompt: the previous value of the in-phase prompt correlator;
- Tup: the upper threshold equal to $25T_{obs}$, where T_{obs} is the observation time. The threshold is set according the fact that 25 bit transitions should occur each second;
- Tlow: the lower threshold;
- Synched: boolean variable telling if bit synchronization has been achieved.

Implement the bit-syncher as a single Matlab structure (use the struct keyword).
The following function needs to be implemented for updating the histogram bit synchronizer:

```
function HS = AddSymbol(HS, prompt)
```

where HS is the bit syncher structure and prompt is a value from the prompt in-phase correlator. Discuss the impact of bit synchronization on signal tracking.

Exercise No. 2: Bandwidth and Loop Parameters

Change the bandwidth and the order of PLL and DLL and analyze the correlator outputs stored in the file produced by myTracking.m. What is the impact of the loop bandwidth on the PLL/DLL transient time? Disable the bit synchronizer and process the data file using a 1 ms integration. Consider the following bandwidths

$$B_n = [5, 7.5, 10, 15] \text{ Hz}$$

and estimate the variance of the corresponding Doppler estimates stored in the output file. Plot the Doppler variance as a function of B_n. Comment on the relationship between the Doppler variance and the loop bandwidth.

Exercise No. 3: C/N_0 Estimator

In order to measure the signal quality, implement the Van Dierendonck C/N_0 estimator. In this particular implementation, the average of the ratio between narrow and wide power can be estimated using an exponential filter controlled by the forgetting factor α. More specifically,

$$\text{WBP} = \sum_{i=1}^{20} \left[p_{I,i}^2 + p_{Q,i}^2 \right] \tag{1}$$

$$\text{NBP} = \left[\sum_{i=1}^{20} p_{I,i} \right]^2 + \left[\sum_{i=1}^{20} p_{Q,i} \right]^2 \tag{2}$$

$$\text{NP} = \frac{\text{NBP}}{\text{WBP}} \tag{3}$$

$$\mu_h = \alpha \mu_{h-1} + (1 - \alpha) NP. \tag{4}$$

The estimator can be implemented using a single Matlab structure with the following fields:

- complexCorr: a vector containing 20 complex prompt correlators belonging to the same data bit;
- index: pointer to the current complex correlator. complexCorr and index implement a circular buffer;
- alpha: the forgetting factor of the exponential filter;
- mu: the average of NP. mu should be initialized to zero and updated using (4) every time a new estimate of NP is available;
- cno the current estimate of the C/N_0.

The C/N_0 should be updated using the following function

CNoEstimator = UpdateCNoEstimator(CNoEstimator, prompt)

where 'prompt' is a complex correlator integrated over 1 ms. UpdateCNoEstimator should be called only after achieving bit synchronization and should implement the following operations:

- store the complex correlators into the circular buffer defined by complexCorr and index;
- evaluate the new C/N_0 estimate each time 20 correlators have been stored.

The C/N_0 estimates should be stored into the vector cnoValue and saved to file.