Receiver Technology

Homework 1a:Signal Acquisition

Marina Todorovska
ESPACE spetialization:Navigation

Introduction

GPS received signals cannot be directly used by the end user. They first need to be processed because of various reasons. The arriving signal power is very low, and needs to be increased. Along with the desired signals come replicas if multipath is present, and the always existing noise. These nuisance components need to be removed to the biggest extent possible. GPS carrier frequency is around 1.5 GHz, which demands powerful computers for processing. Instead of providing very powerful computers, the frequency is downconverted. The received signal is analog. An A/D (analog-to-digital) converter must be used, usually in GPS 2 bit A/D converter with 4 levels of quantization. The chosen sampling rate has to be high enough so that the original signal can be reconstructed.

In the first task a dummy C/A code is generated, using two maximum length shift registers of 10 registers with a length of 1023. Within the same family of codes, 32 Gold sequences are generated for 32 GPS satellites.

The reason Gold sequences are used, are their high auto-correlation properties (to measure similarity when matching the signal), and the low cross-correlation properties (for channel sharing by different GPS satellites).

In the last part of the task power spectral density (PSD) is analyzed.

Implementation of a sampled C/A code generator with a given code delay

Generation of Gold code sequence

C/A code is a pseudorandom code from the family of Gold code. It is periodic with 1023 chips pattern of repetition. The duration of a chip is 1us, which means that in one second there are 1.023 Mcps.

Gold codes are constructed for $N=2^n-1$, for $n \neq 0 \pmod 4$. For the case of GPS C/A code bit values of 1 and -1 are used.

In Table 1 the combinations of maximum shift length registers used are represented. Two maximum length shift registers are used (MLS1 and MLS2), each with a length of 10. From each register output bits are taken, which are then used as inputs to a modulo 2 operation (same bits output -1, different bits output 1). This is illustrated in Figure 1, as bits 1,3,10 from MLS1 and 1,2,3,6,8,9,10 from MLS2.

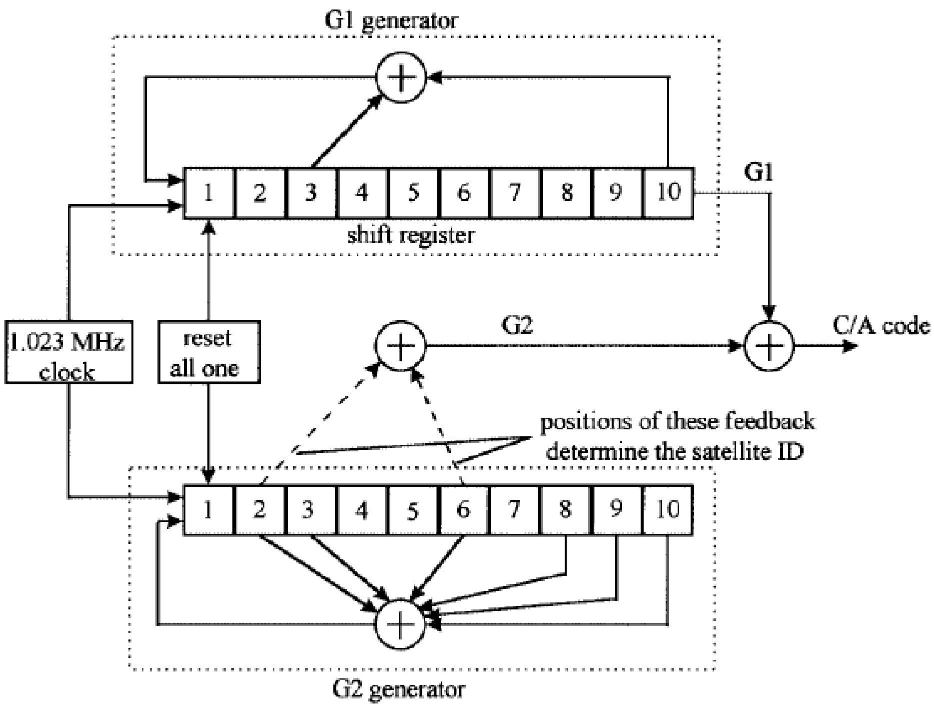

Figure 1 C/A code generation

There are 32 satellites in orbit at the moment. For PRN of 1 to 32 each generated C/A code is unique. For PRN above 32 to 37 the same C/A code is used. The third column in Table 1 named code phase selection shows for each satellite which register is used. For example for PRN 8 bits 2 and 9 from MLS2 form a delay of 140 chips in the code. The fifth column when converted to binary shows what the expected match should be.

Table 1 C/A code phase assignments

Satellite ID Number	GPS PRN Signal Number	Code Phase Selection	Code Delay Chips	First 10 Chips C/A Octal
1	1	$2 \oplus 6$	5	1440
2	2	$3 \oplus 7$	6	1620
3	3	$4 \oplus 8$	7	1710
4	4	$5 \oplus 9$	8	1744
5	5	$1 \oplus 9$	17	1133
6	6	$2 \oplus 10$	18	1455
7	7	$1 \oplus 8$	139	1131
8	8	$2 \oplus 9$	140	1454
9	9	$3 \oplus 10$	141	1626
10	10	$2 \oplus 3$	251	1504
11	11	$3 \oplus 4$	252	1642
12	12	$5 \oplus 6$	254	1750
13	13	$6 \oplus 7$	255	1764
14	14	$7 \oplus 8$	256	1772
15	15	$8 \oplus 9$	257	1775
16	16	$9 \oplus 10$	258	1776
17	17	$1 \oplus 4$	469	1156
18	18	$2 \oplus 5$	470	1467
19	19	$3 \oplus 6$	471	1633
20	20	$4 \oplus 7$	472	1715
21	21	$5 \oplus 8$	473	1746
22	22	$6 \oplus 9$	474	1763
23	23	$1 \oplus 3$	509	1063
24	24	$4 \oplus 6$	512	1706
25	25	$5 \oplus 7$	513	1743
26	26	$6 \oplus 8$	514	1761
27	27	$7 \oplus 9$	515	1770
28	28	$8 \oplus 10$	516	1774
29	29	$1 \oplus 6$	859	1127
30	30	$2 \oplus 7$	860	1453
31	31	$3 \oplus 8$	861	1625
32	32	$4 \oplus 9$	862	1712
**	33	$5 \oplus 10$	863	1745
**	34*	$4 \oplus 10$	950	1713
**	35	$1 \oplus 7$	947	1134
**	36	$2 \oplus 8$	948	1456
**	37*	$4 \oplus 10$	950	1713

Generation of sampled C/A code

Sampling rate of 1/5.714 $*10^6$ s is used. The duration of the code is 1 ms , and the sampling frequency becomes 5,714 MHz. In 1ms now there are 5714 bit, spreaded from the original sequence of 1023 which is obtained in the previous task. Basically the chip sequence is divided further into smaller lengths of 1/5,714 us, such that to each length a bit is assigned that matches the value of the corresponding 1us chip to which it belongs to originally. Ultimately there are approximately 5 to 6 bits spreading.

Generation of a shifted and sampled C/A code

Upon arrival of the signal to the receiver there is certainly some delay with respect to the sent signal. The C/A code as we saw is periodically repeating, so any delay will simply result in circular shifting of the original code, thus not leading to information loss.

The number of the sampled bits which correspond to the chip delay is the number of sampled bits per chip.

Correlation properties of the C/A codes

Auto correlation function measures the similarity of two sequences when one is a time shift of the other. Cross correlation is used to compare a sequence with all the time shifts of another sequence. A function with good cross-correlation properties can be used to detect a weak signal in the presence of strong signals. That means that signals from different GPS satellites can be easily distinguished, thus no offset in the transmission frequency is required. GPS uses CDMA (code division multiple access). Signals have slightly different frequencies because of the different Doppler shifts.

In Figure 2 the correlation properties of PRN1 and PRN2 are shown.

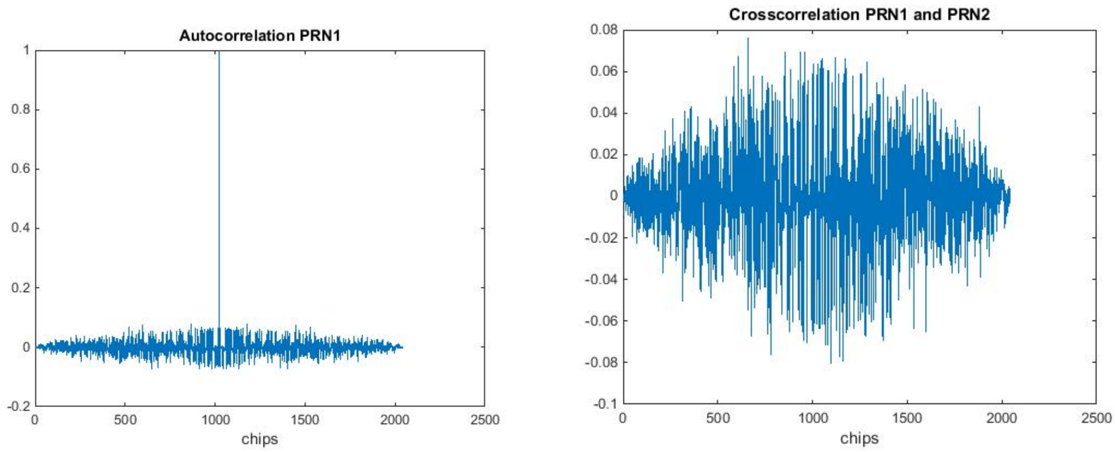

Figure 2 Auto-correlation of PRN 1 and Cross-correlation of PRN 1 and PRN 2

The autocorrelation peak is at 1023 (length of C/A code), while the rest of the sequence differs by small values from zero due to the Gold codes being not completely orthogonal.

The cross correlation is takes values not bigger than +/- 0.08. It is very noisy, so it is not clearly distinuishable. From the Gold code theory, the peaks of a cross correlation of two Gold codes should have peaks at values of 0.062, 0.001 and

0.064. This is calculated from $\left\{-\frac{1}{N}, -\frac{\beta(n)}{N}, \frac{\beta(n)-2}{N}\right\}$, where $\beta(n) = 1 + 2^{\frac{n+2}{2}}$, with n=10 e.g. the number of shift registers, N=1023, the length of the sequence.

In Figure 3 the sampled correlation functions are plotted.

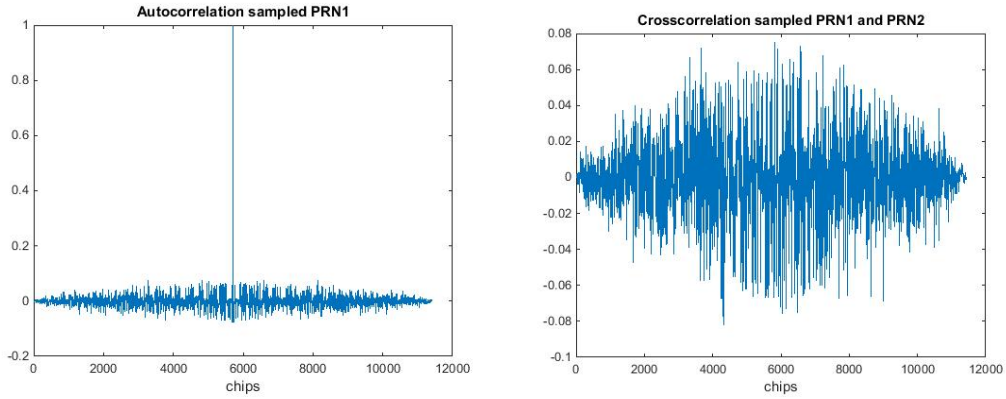

Figure 3 Auto-correlation of PRN 1 and Cross-correlation of PRN 1 and PRN 2

Here the autocorrelation peak is at 5714, corresponding to the length of the spreaded code. The cross-cprrelation is spreaded across 5714*2-1 chips, and again is not very distinguishable. However, the values are again not bigger then +/-0.08, as expected.

Incoming IF file

The given IncomingIF.mat file was read and its time series for the first millisecond is plotted in Figure 4.

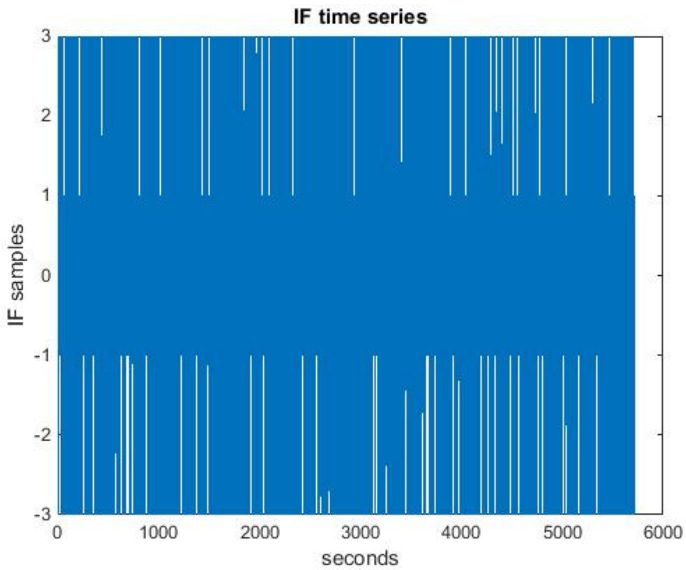

Figure 4 IF time series first milisecond

In Figure 5 the Power Spectral Density is plotted. Power Spectral Density describes how power of a signal or time series is distributed over frequency. To calculate it, the Fourier transform of the correlation function for the positive frequencies is shown (negative spectrum is symmetric). At frequency of 1.5 MHz the strongest peak is noticed. All the values depend on the magnitude and samples of the Incoming IF time series.

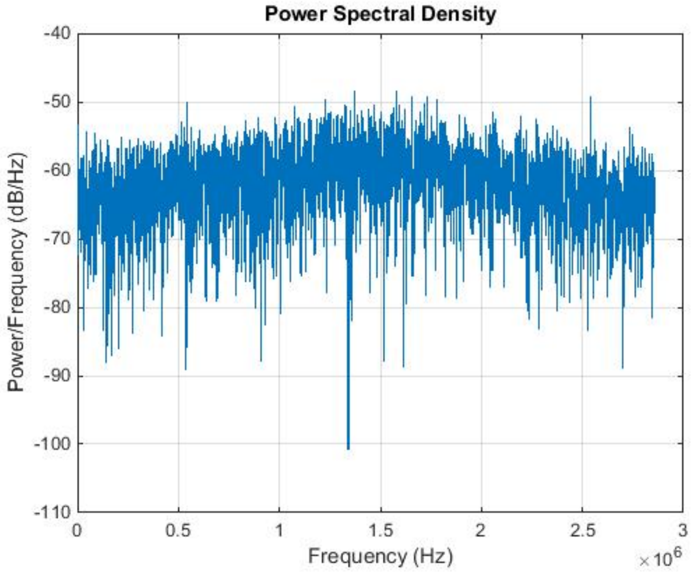

Figure 5 Power Spectral Density

Code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Homework 1a
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [ code ] = CA( PRN )

if (PRN>32)||(PRN<1) % Error checking
 error('Enter satellite number between 1 and 32\n')
end
integer=round(PRN)-PRN;
if integer ~= 0
 warning('Satellite number is integer');
 PRN = round(PRN);
end

% table of C/A Code Tap Selection (sets delay for MLS2 generator)
tap=[2 6; 3 7; 4 8; 5 9; 1 9; 2 10; 1 8; 2 9; 3 10; 2 3; 3 4; 5 6; 6 7;
7 8; 8 9; 9 10; 1 4; 2 5; 3 6; 4 7; 5 8; 6 9; 1 3; 4 6; 5 7; 6 8; 7 9; 8
10; 1 6; 2 7; 3 8; 4 9];

n=10; % Registers
L=2^n-1; % PRN sequence length
s=[0 0 1 0 0 0 0 0 0 1]; % Register MLS1 taps
t=[0 1 1 0 0 1 0 1 1 1]; % Register MLS2 possible taps
g1=ones(1,n); % initialize MLS1
q=ones(1,n); % initialize MLS2

%generation C/A code sequence
tapp=tap(PRN,:);
for step=1:L % for each bit
 g2(:,step)=mod(sum(q(tapp),2),2); % mod-2 sum tap points
```

```
 code(:,step)=mod(g1(n)+g2(:,step),2); % code bit value is mod-2
% of g1 and g2
 g1=[mod(sum(g1.*s),2) g1(1:n-1)]; % shift MLS1
 q=[mod(sum(q.*t),2) q(1:n-1)]; % shift MLS2
end
for step=1:L % Gold code
 if code(step)==0
    code(step)=-1;

 end
code=code';
end

%Sampled code C/A
function sampled_code=SampledCA(PRN,Ts)

code=(CA(PRN))';
%1 s is 1000ms which is the duration of a sequence in C/A code
b=(10^-3)/Ts;
%one chip period is
Ts=Ts*10^6;
codesampled=[];
a=1;

for i=1:1023
    j=a;
    for j=a:b
            sum=Ts*j;
            if sum <i
            codesampled=horzcat(codesampled,code(1,i));
            a=a+1;
            else
            codesampled=horzcat(codesampled,code(1,i+1));
            a=a+1;
            break
        end
     end
end
sampled_code=codesampled;
end

%Shift the code circulary for a given delay

shifted_sampled_code=sampled_code;
delay=ceil(code_delay*1/Ts); % [chip]*[sample bit/chip]=[sample bits]
i=1;
for count=delay:5714
shifted_sampled_code(i)=codesampled(count);
i=i+1;
end

for count=1:delay-1
shifted_sampled_code(i)=codesampled(count);
i=i+1;
end

shifted_sampled_code=codesampled;
end

%Generate the auto and cross correlation plots
```

```
ca1=CA(1);
ca2=CA(2);
Ts=1/5714000;
sampled_code1=SampledCA(1,Ts);
sampled_code2=SampledCA(2,Ts);


figure(1)
plot(xcorr(ca1)/length(ca1));
xlabel('chips')
title('Autocorrelation PRN1')
figure(2)
plot(xcorr(ca1,ca2)/length(ca1));
xlabel('chips')
title('Crosscorrelation PRN1 and PRN2')
figure(3)
plot(xcorr(sampled_code1)/length(sampled_code1));
xlabel('chips')
title('Autocorrelation sampled PRN1')
figure(4)
plot(xcorr(sampled_code1,sampled_code2)/length(sampled_code1));
xlabel('chips')
title('Crosscorrelation sampled PRN1 and PRN2')

%Plot the IF file and PSD

b=struct2cell(a);
c=cell2mat(b);
incoming_1ms_IF=c(1,:);

figure(1)
plot(1:5714,incoming_1ms_IF);
xlabel('seconds')
ylabel('IF samples')
title('IF time series')

figure(2)
Fs = 5714000;
t = 0:1/Fs:1-1/Fs;
N = length(incoming_1ms_IF);
xdft = fft(incoming_1ms_IF);
xdft = xdft(1:N/2+1);
psdx = (1/(Fs*N)) * abs(xdft).^2;
psdx(2:end-1) = 2*psdx(2:end-1);
freq = 0:Fs/length(incoming_1ms_IF):Fs/2;
plot(freq,10*log10(psdx))
grid on
title('Power Spectral Density')
xlabel('Frequency (Hz)')
ylabel('Power/Frequency (dB/Hz)')
```