ASEN 5090 - Intro to GNSS
Homework 9

Jake Vendl and Jack Toland

Problem 1 - Visibility Prediction

Using code from homework 3, the visibile satellites on August 28th, 2018 at 16:29 UTC were found. The resulting sky plot is shown below:

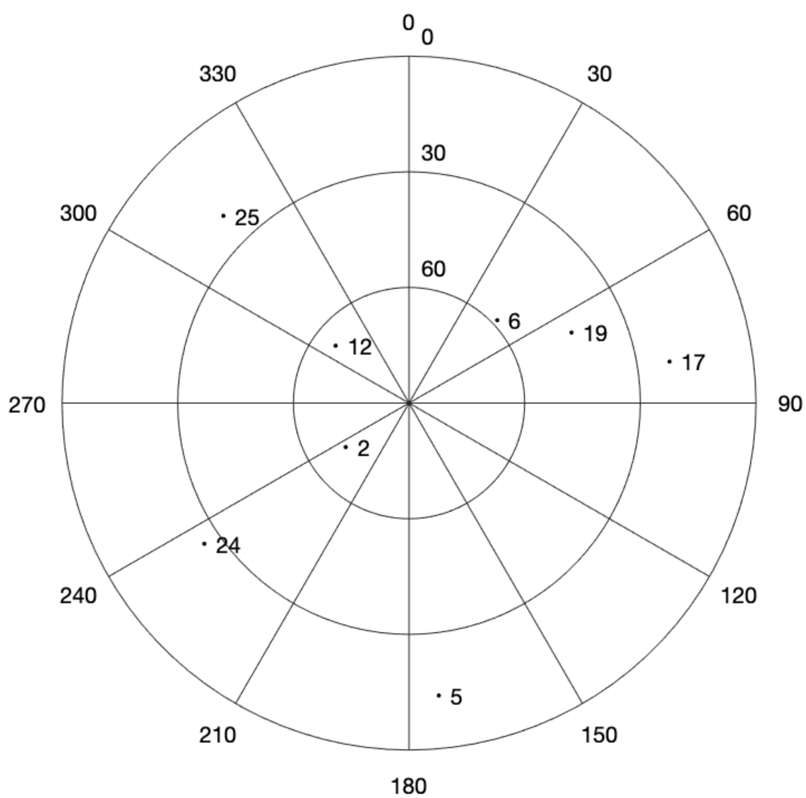

Figure 1: Skyplot for problem 1, showing PRNs 2 and 12 at greater than 60deg visibility

Problem 2 - Carrier Wipeoff and Code Correlation

This part of the assignment calls for the creation and execution of a complex correlator which is able to process provided data. This was accomplished using the following equation, also provided in the assignment document.

$$S(f_d, \tau) = \sum_{n=1}^{n} s_R(i + \frac{\tau}{\Delta t_s}) x_i e^{-j\theta_i} \tag{1}$$

To enact this, we coded up a loop that iteratively adds to a running sum, calling on parts of the provided data at different points. The functionality of this code block was checked using the provided examples in the assignment document.

```
PRN 2, 1ms Integration Time
DOP = 0 kHz  DELAY = 9 samples  S = -66.4585+100.0478i
DOP = 1000 kHz  DELAY = 2943 samples  S = 1597.8253+-390.3275i
```

Problem 3 - Search Grid

Conflict arose in this section of the assignment. To start, it was decided that the delay axis would run from 1 to 6626, representing one millisecond with increments of a step size corresponding to the sampling frequency. For the Doppler axis, a grid from -5000Hz to +5000Hz was chosen so that almost all possible delays would be captured in the search grid. Then, a series of loops were woven together to sew up a bunch of complex correlator outputs, essentially $S(f_d, \tau)$ for a range of f_d and τ. Then, the plots were examined to find peaks, with the peaks representing the correct Doppler shift and chip delay. For PRN 2, the results were the following: Doppler shift of 1kHz and chip delay of 2943 (≈863,000 meters).

Shown below is the 3D mesh generated for PRN 2, with the complex correlator output shown as a function of Doppler shift and chip delay.

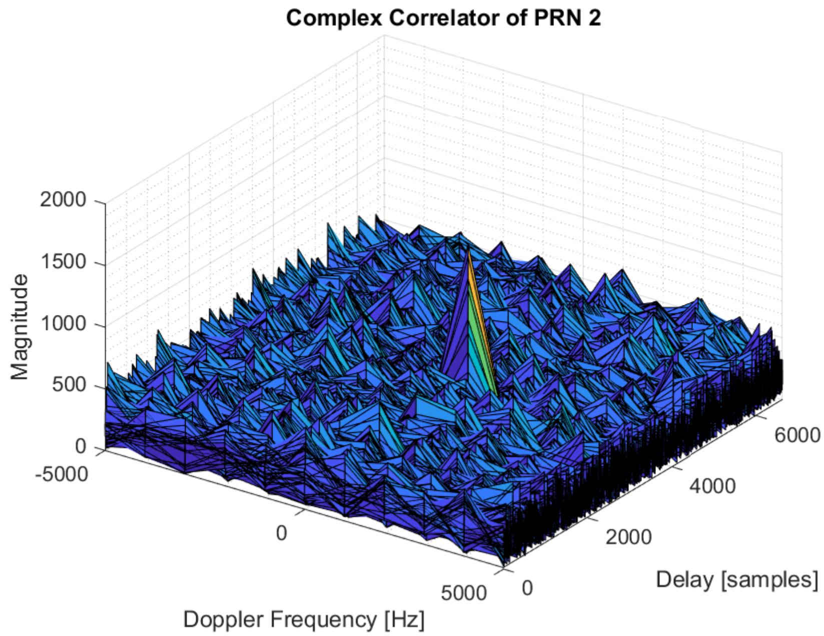

Figure 2: PRN 2 Complex Correlator - 1ms integration time

Shown below are both of the cross-sectional plots of the above grid. First, S as a function of chip delay, shown from the peak Doppler bin.

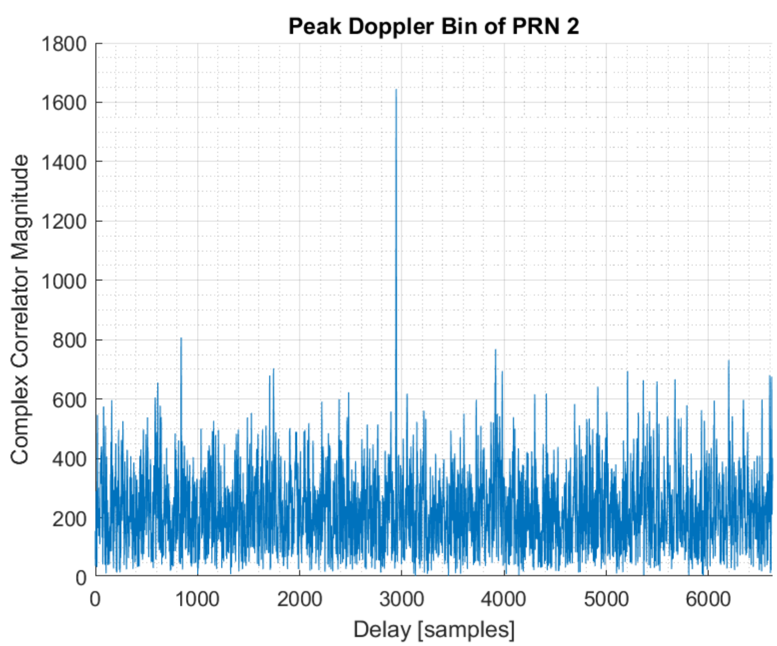

Figure 3: Chip delay of PRN 2, cross-section taken from f_d=1kHz

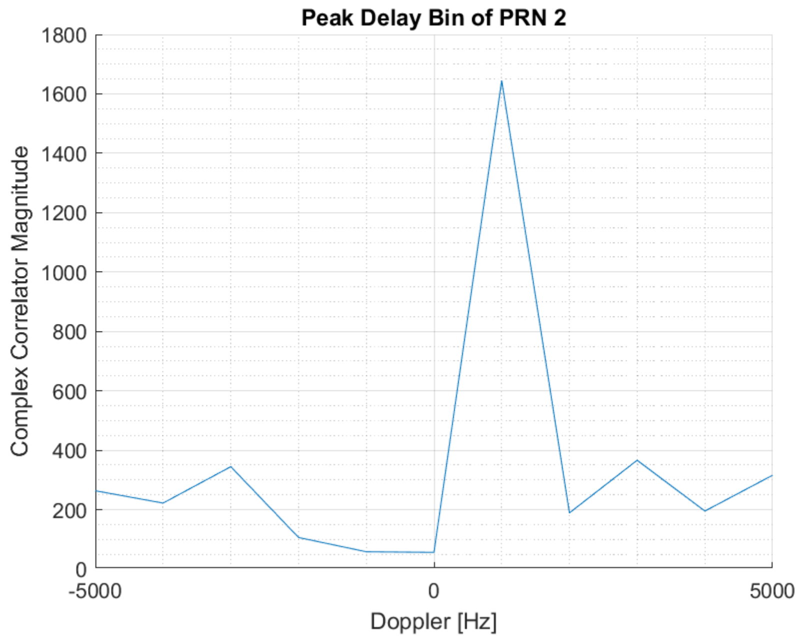

Figure 4: Doppler shift of PRN 2, cross-section from τ=2943 chips

Problem 4 - Find Additional Satellites

The three additional satellites are PRNs 5, 6, and 12, all chosen because they are visible at this moment in time. PRN 12 is at a high elevation, 6 is a midrange elevation, and PRN 5 is at a low elevation.

0.1 PRN 5

First, the plots for PRN 5. The peak for this PRN is at a Doppler shift of 4kHz and a chip delay of 4582 samples ($\approx$1,344,000 meters).

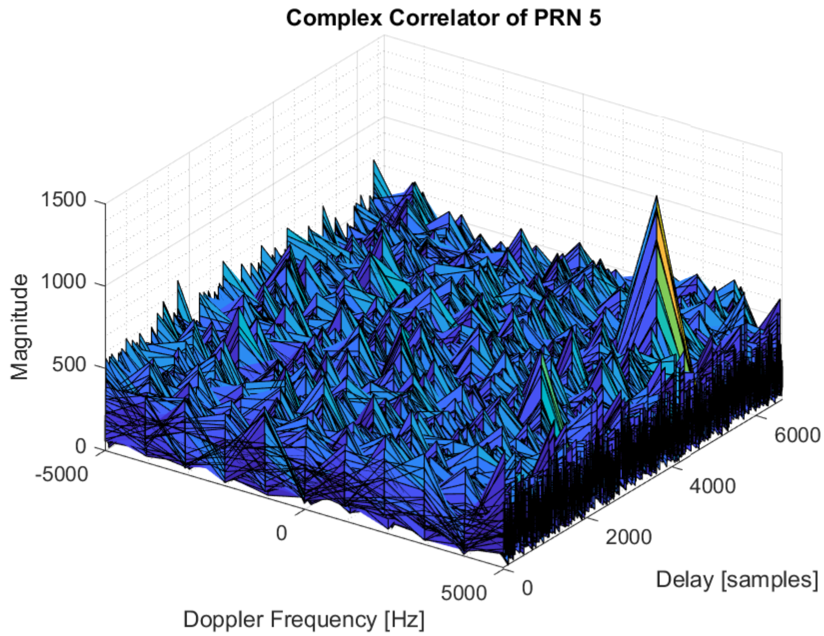

Figure 5: PRN 5 Complex Correlator - 1ms integration time

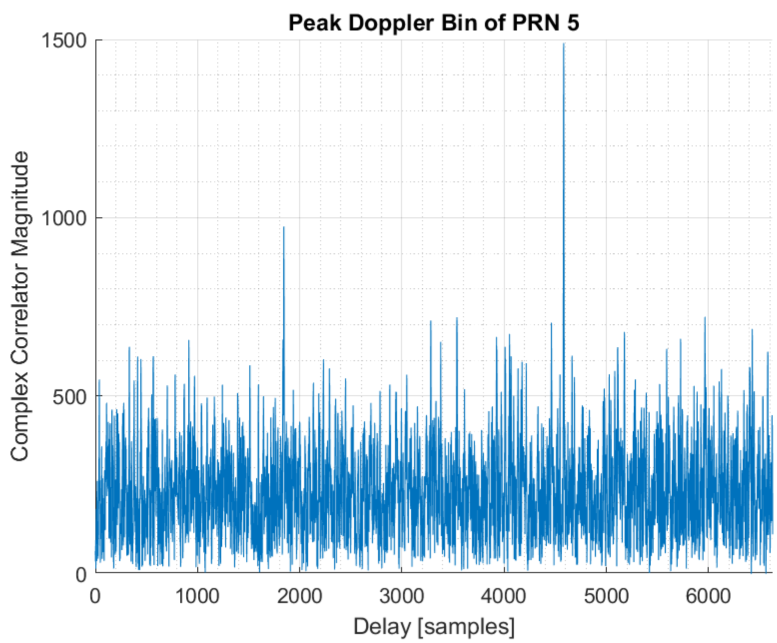

Figure 6: Chip delay of PRN 5, cross-section taken from f_d=4kHz

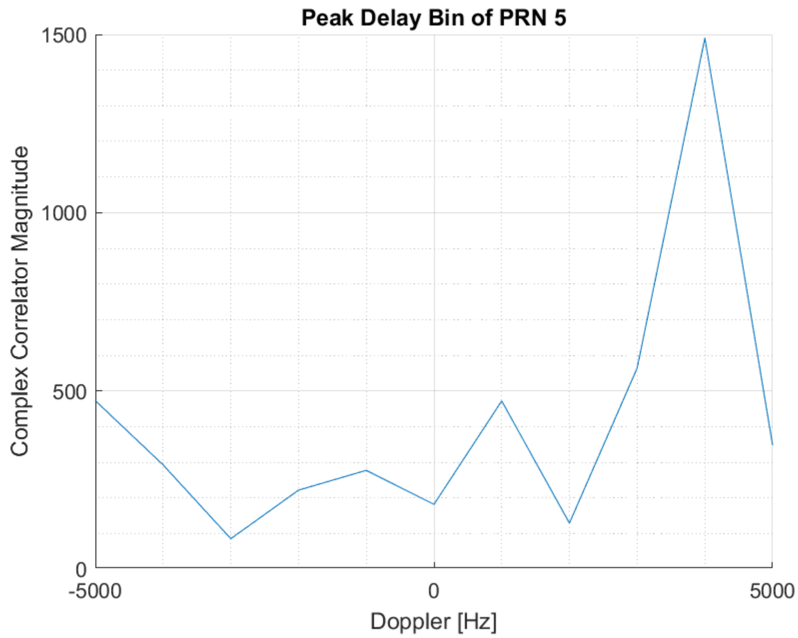

Figure 7: Doppler shift of PRN 5, cross-section from τ=4582 chips

0.2 PRN 6

Next, the plots for PRN 6, which has a peak at a Doppler shift of -2kHz and a chip delay of 2556 samples (≈750,000 meters).

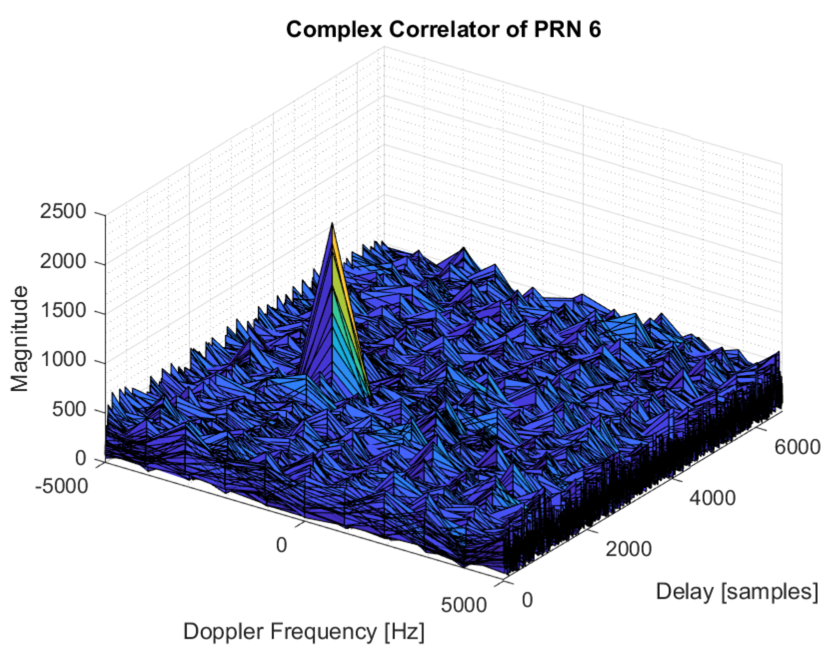

Figure 8: PRN 6 Complex Correlator - 1ms integration time

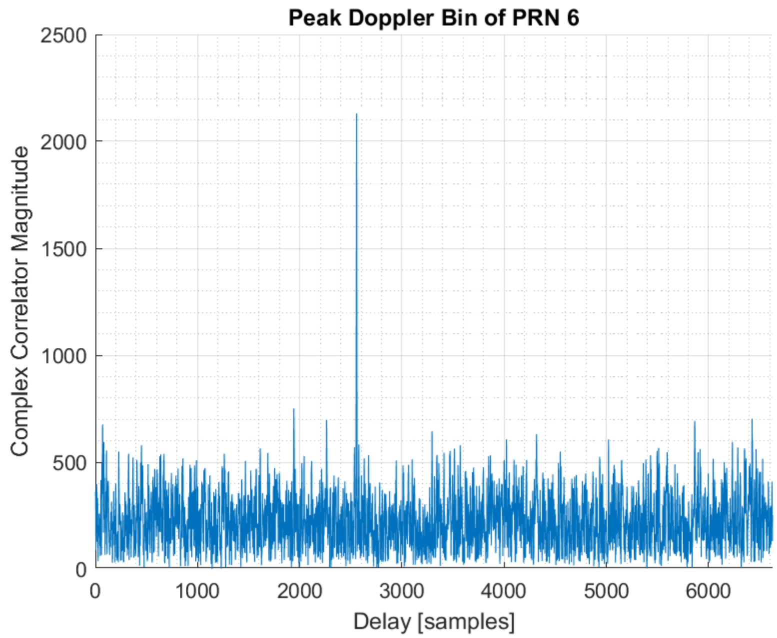

Figure 9: Chip delay of PRN 6, cross-section taken from f_d=-2kHz

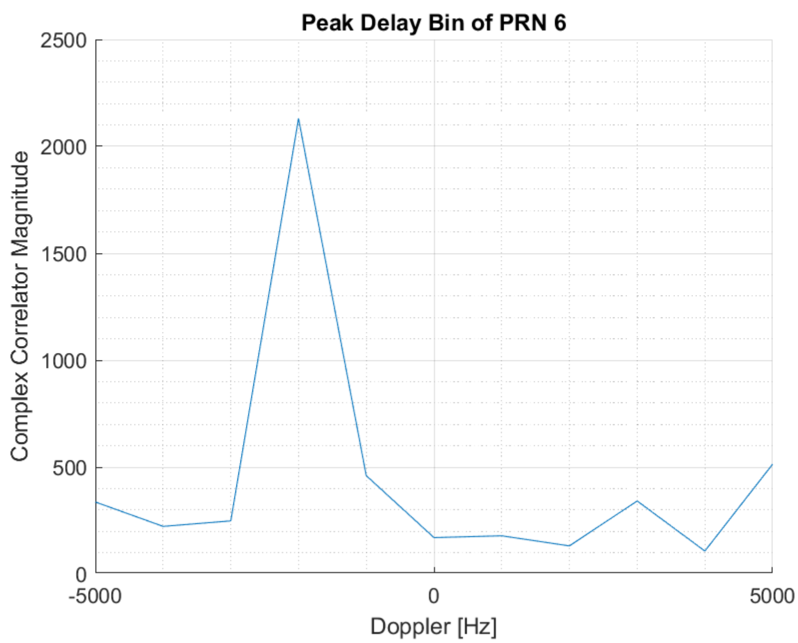

Figure 10: Doppler shift of PRN 6, cross-section from τ=2556 chips

0.3 PRN 12

Next, the plots for PRN 12, which has a peak at a Doppler shift of 1kHz and a chip delay of 6541 samples (≈1,918,000 meters).

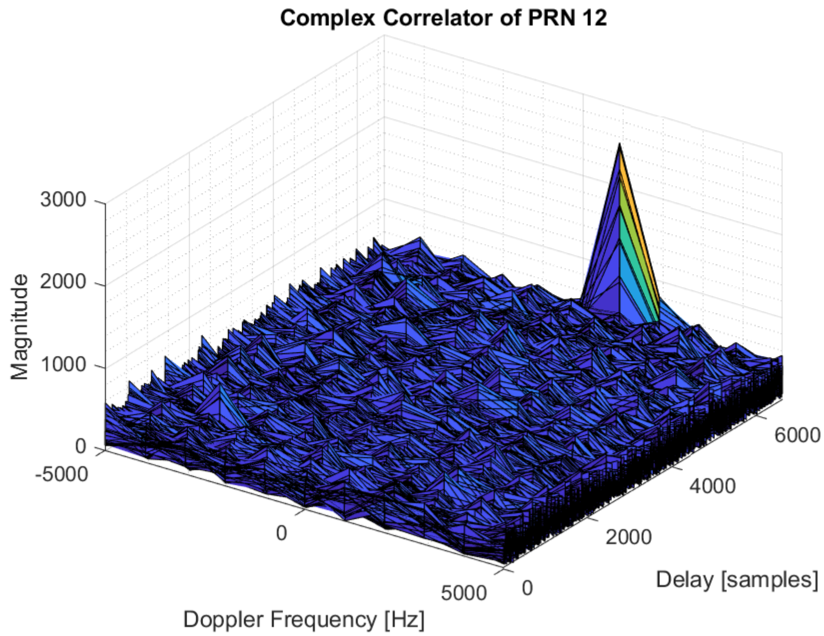

Figure 11: PRN 12 Complex Correlator - 1ms integration time

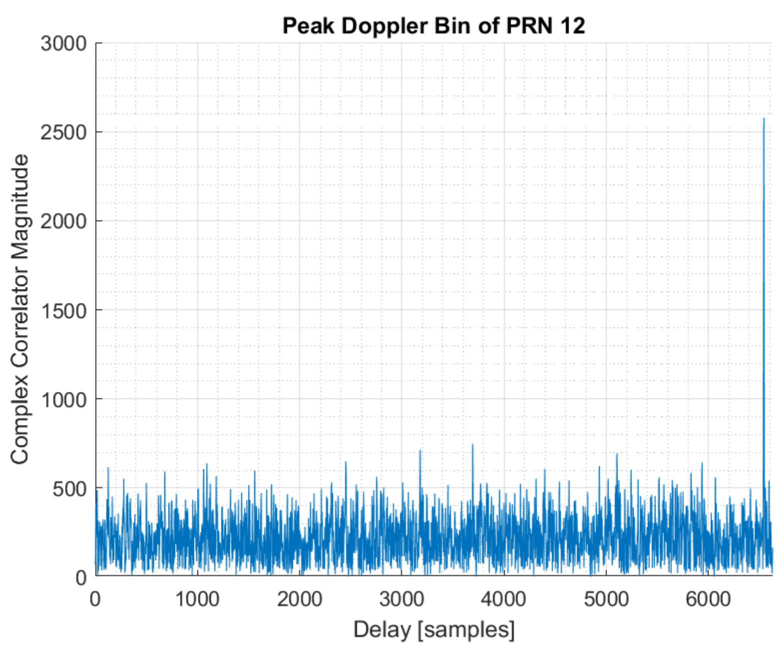

Figure 12: Chip delay of PRN 12, cross-section taken from f_d=1kHz

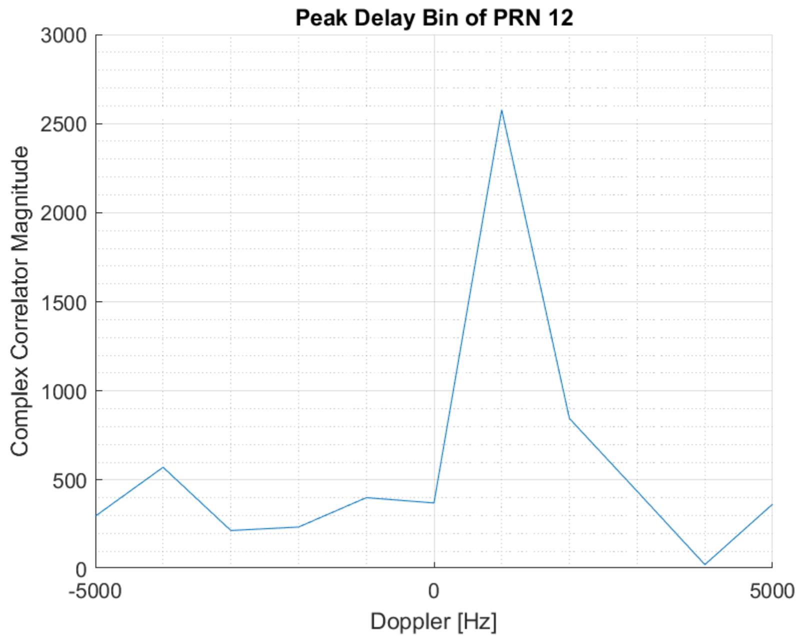

Figure 13: Doppler shift of PRN 12, cross-section from τ=6541 chips

Problem 5 - Increase Integration Time to 2ms

Increasing the integration time to 2ms, effectively enabling the complex correlator function to consider more of the signal data. This has the expected effect of increasing correlation peaks, as shown below.

0.4 PRN 5

First, the plots for PRN 5. The peak for this PRN is at a Doppler shift of 3.5kHz and a chip delay of 4582 samples (≈1,344,000 meters).

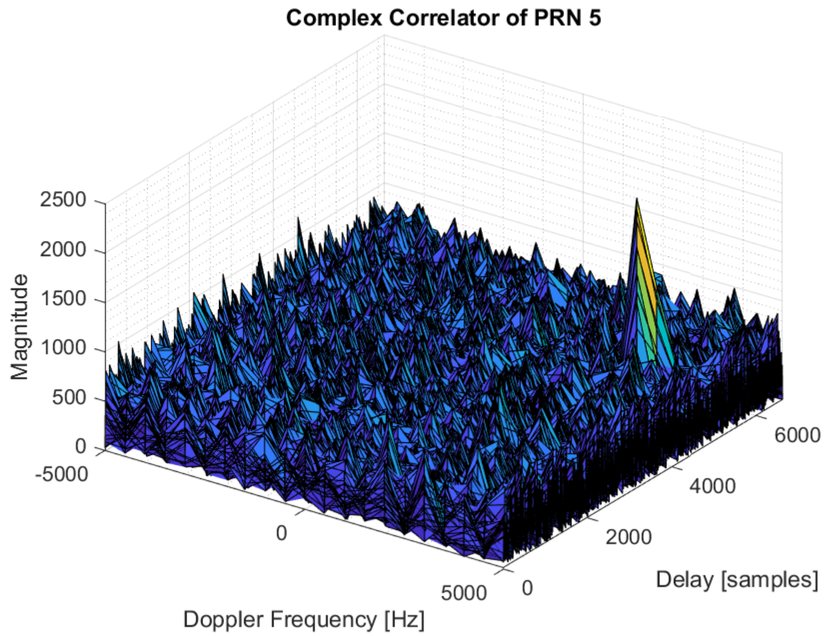

Figure 14: PRN 5 Complex Correlator - 2ms integration time

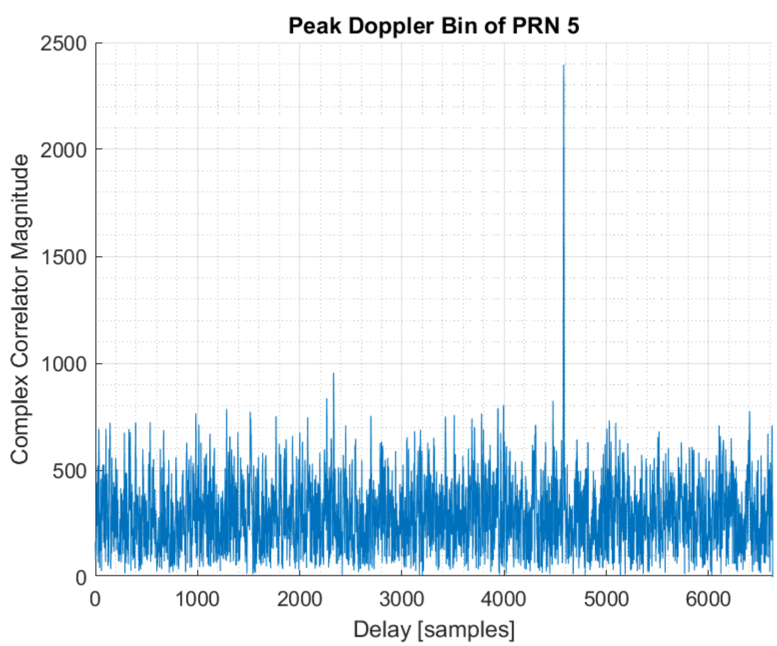

Figure 15: Chip delay of PRN 5, cross-section taken from f_d=3.5kHz

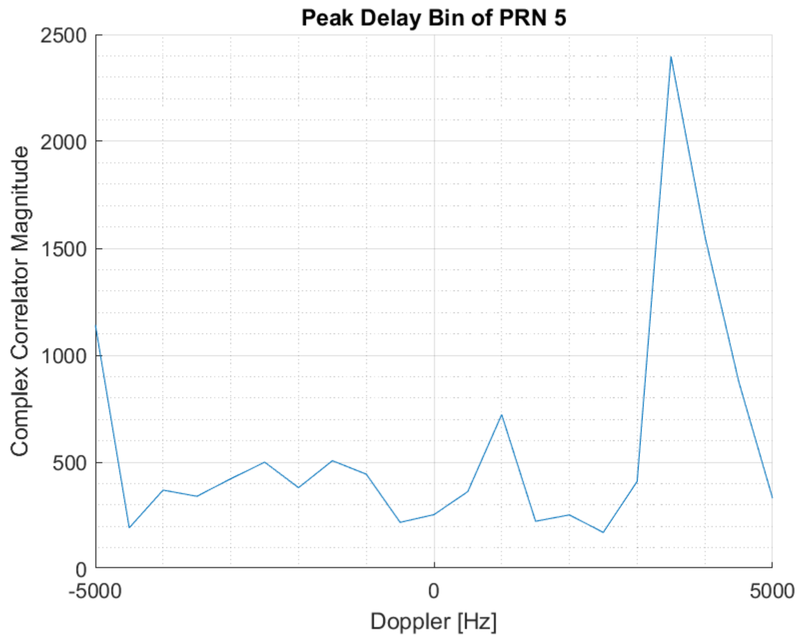

Figure 16: Doppler shift of PRN 5, cross-section from τ=4582 chips

0.5 PRN 6

Next, the plots for PRN 6, which has a peak at a Doppler shift of -2kHz and a chip delay of 2556 samples ($\approx$750,000 meters).

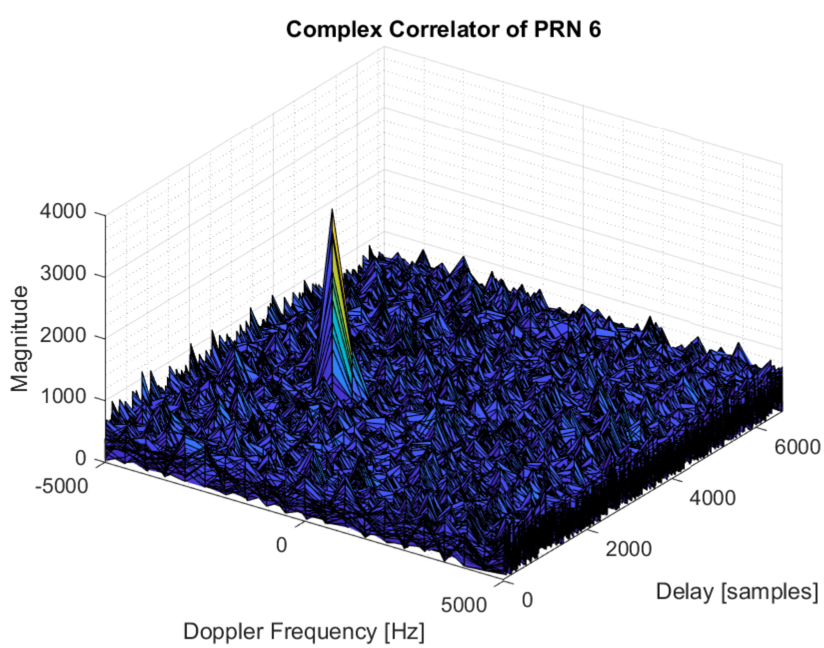

Figure 17: PRN 6 Complex Correlator - 2ms integration time

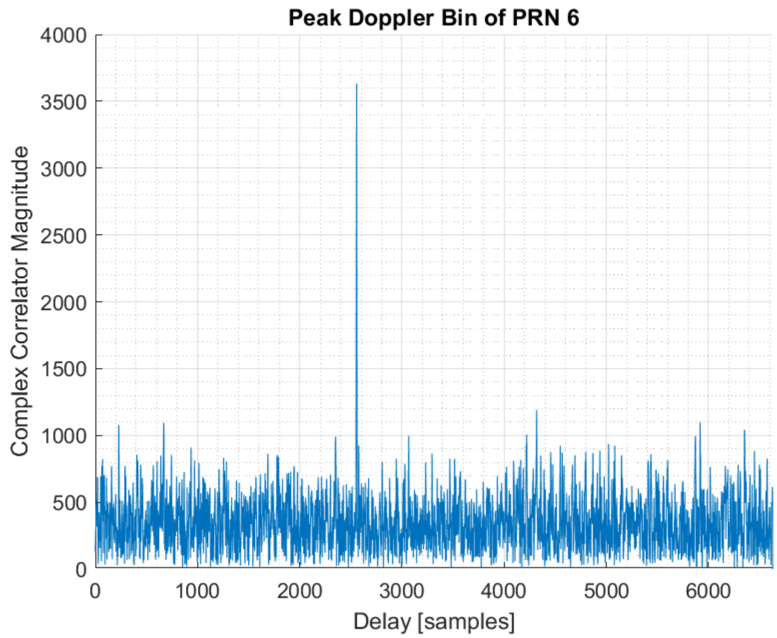

Figure 18: Chip delay of PRN 6, cross-section taken from f_d=-2kHz

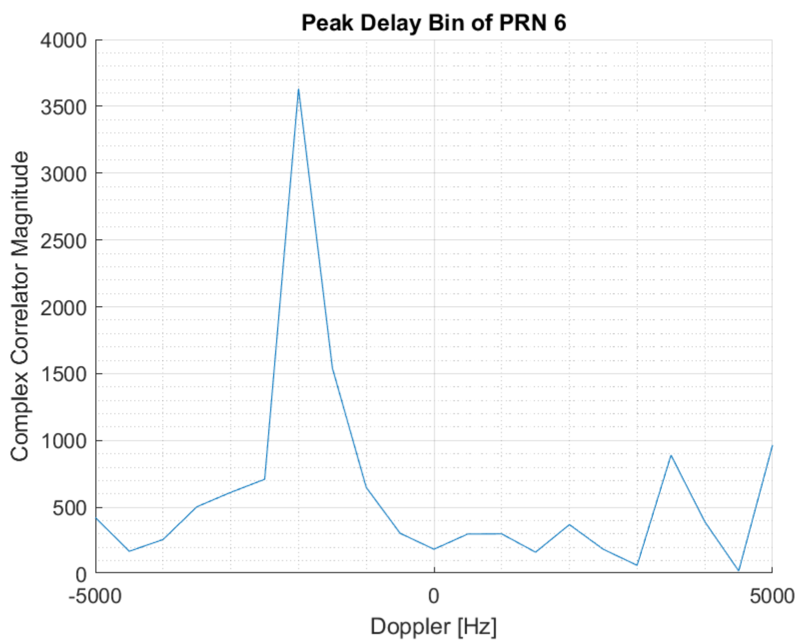

Figure 19: Doppler shift of PRN 6, cross-section from τ=2556 chips

0.6 PRN 12

Next, the plots for PRN 12, which has a peak at a Doppler shift of 1kHz and a chip delay of 6541 samples (≈1,918,000 meters).

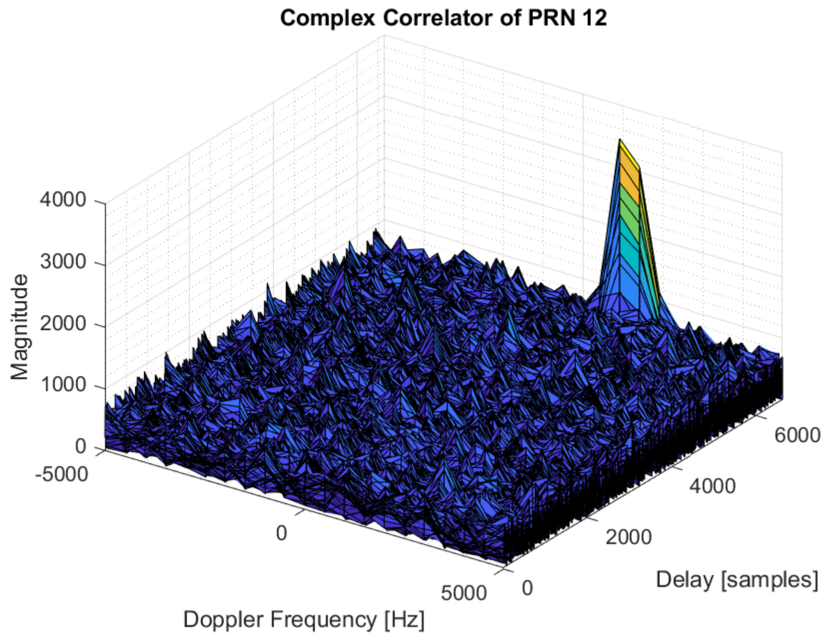

Figure 20: PRN 12 Complex Correlator - 2ms integration time

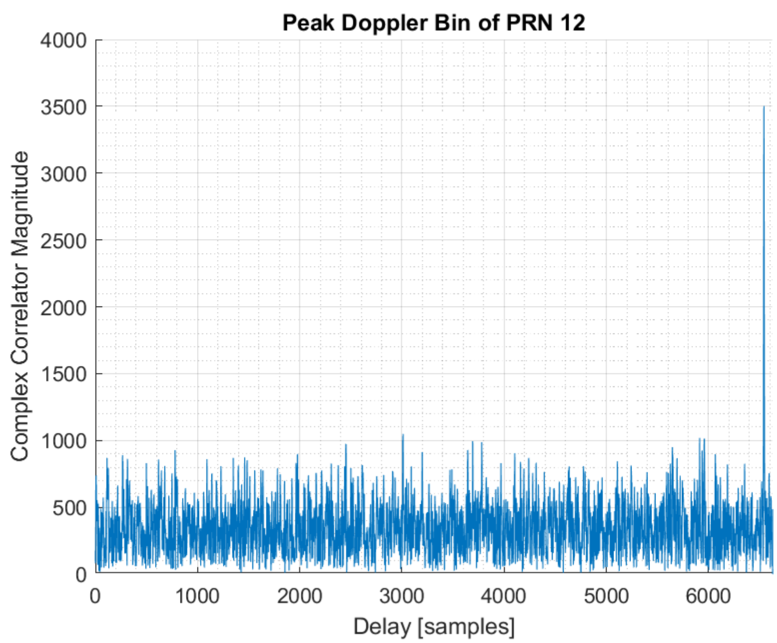

Figure 21: Chip delay of PRN 12, cross-section taken from f_d=1kHz

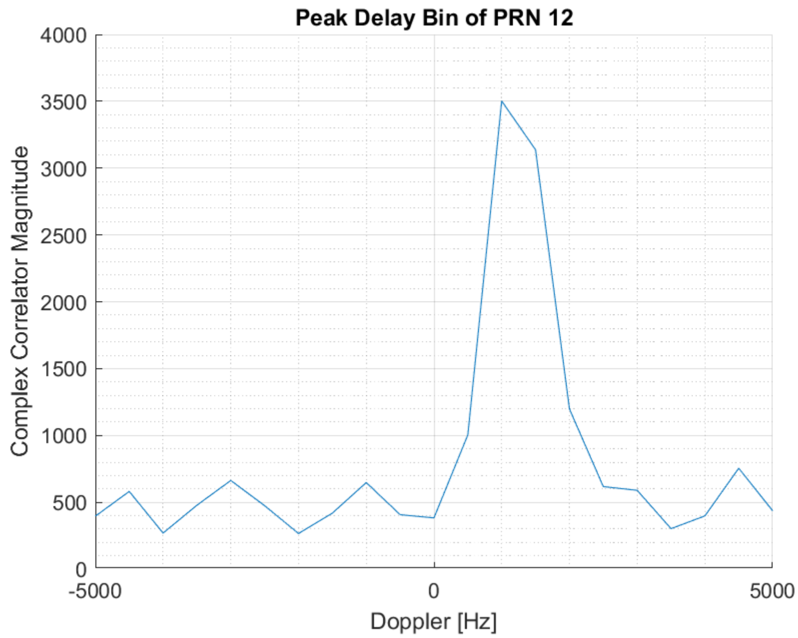

Figure 22: Doppler shift of PRN 12, cross-section from τ=6541 chips

Results

The effect of 2ms integration time is shown below in table format, highlighting the differences in complex correlator peak sizes as a function of integration time. Here, S(f_d,τ) is calculated as the norm of the real and imaginary parts of the complex correlator. As such, it is unitless and only resembles a global maximum of the considered grid.

	S($f_{d,peak}$,τ_{peak})	
PRN	1ms	2ms
5	1489.14	2396.77
6	2130.72	3631.54
12	2577.46	3501.75

As can be seen in the table above, increasing the integration time has an definitive impact on the output of the complex correlator. Intuitively, this makes sense: as the integration time increases to 2ms, two entire loops of the CA code are being considered by the complex correlator, making correlation more definitive.

A future analysis which could be revealing for this data set is to calculate the signal-to-noise ratio of complex correlator peaks as a function of integration time. A hypothesis is that the peaks would grow larger as the noise floor amplitude would stay roughly the same, but a numerical analysis could be performed to confirm these suspicions. This would result in a greater signal to noise ratio (SNR) as the peak magnitude grows relative to the noise magnitude.

This assignment did well to reveal how a GPS receiver begins to acquire a GPS signal. Starting from an incoming stream of data, the complex correlator is able to take the CA code for a given PRN and subsequently determine if that PRN has a match in the incoming data stream. That match, should it be found, will have a corresponding Doppler shift and chip delay which can be passed to a tracking algorithm.

Code

```
%==========================================================================
% Jake Vendl | Jack Toland
% ASEN 5090
% Homework 9
% 12/6/2019
%==========================================================================

clear all; close all; clc

addpath('GPS Functions');

fn = 6.625e6;
fIF = -60e3;

lat=40.0; long = -105.15; alt = 1629;
GPS_LLA = [lat; long; alt]';
GPS_ECEF = lla2ecef(GPS_LLA);

%% =======================================================================
% Problem 1 - Visiblity Prediction
%==========================================================================
% Data collected on August 28th, 2018 at 16:29 UTC
yumafilename = 'YUMA240.ALM';
[gps_ephem,gps_ephem_cell] = read_GPSyuma(yumafilename);
week = cal2gps([2018, 08, 28]);
tow = [week, 2*86400+(16*60+29)*60]; %(16 hrs times 60 m/hr + 29min)*60s/m

for i=1:size(gps_ephem,1)
    PRN(i) = gps_ephem(i,1);
    [~, pos] = broadcast2pos(gps_ephem, tow, PRN(i));
    satECEF(i,1:3) = pos';
    clear pos
end

% Find az el of all sats at 16:29
count=1;
for i=1:size(satECEF,1)
    [az,el,~] = compute_azelrange(GPS_ECEF,satECEF(i,:));
    if el > 0
        azimuth(count) = az;
        elevation(count) = el;
        svs(count) = PRN(i);
        count=count+1;
```

```
    end
end
plotAzEl(azimuth,elevation,svs)


%% ======================================================================
% Problem 2 - Carrier Wipeoff
%=======================================================================

% Load the datafile
load('ASEN5091data.mat');

% Create a time vector at intervals of deltaTs
tstep_sam = 1/fn;
int_time = 0.001; % 1ms
t_vec = 0 : tstep_sam : int_time;

% Create a vector of PRN2 C/A code values
CA_2 = generate_CA_code(2,0.001); %PRN 2 and integration time 1ms

% Match C/A code to time vector
tstep = int_time/length(CA_2);
sig_CA_2 = zeros(1,length(t_vec));
for n = 1:length(t_vec)
    tval = t_vec(n);
    partial_index = tval/tstep;
    index = floor(partial_index)+1;
    if index > length(CA_2)
        index = index - length(CA_2);
    end
    sig_CA_2(n) = CA_2(index);
end

% Example 1
tau=9;
fD = 0;
carrier_phase = 2*pi*(fIF + fD)*t_vec;

S=0;
for i=1:length(t_vec)
    ind = round(i+tau);
    S = S + data(ind)*sig_CA_2(i)*exp(-(1i)*carrier_phase(i));
end
fprintf('DOP = %0.0f kHz  DELAY = %0.0f samples  S = %0.4f+%0.4fi \n',fD,tau,real(S),imag(S));


% Example 2
tau=2943;
fD = 1000;
carrier_phase = 2*pi*(fIF + fD)*t_vec;
```

```
S=0;
for i=1:length(t_vec)
    ind = round(i+tau);
    S = S + data(ind)*sig_CA_2(i)*exp(-(1i)*carrier_phase(i));
end
fprintf('DOP = %0.0f kHz  DELAY = %0.0f samples  S = %0.4f+%0.4fi \n',fD,tau,real(S),imag(S));

%% =========================================================================
% Problem 3 - Create a search grid
%==========================================================================
% Setup the delay axis

show_plot = true;

% Compute and display a 3D mesh
fprintf('Example Peak with Integration Time 0.001 seconds:\n');
[tau, doppler, S_max] = complex_correlator(2,data,t_vec,int_time,show_plot);
fprintf('    PRN %0.0f || Tau = %0.0f samples or %0.2f meters || Doppler = %0.0f Hz || S = %0.4f\n',
    2,tau,(tau/1023*0.001*3e8),doppler,S_max);




%% =========================================================================
% Problem 4 - Find more satellites
%==========================================================================
sats = [5,6,12];
fprintf('Peaks with Integration Time 0.001 seconds:\n');
for s = 1:length(sats) % Look for all satellites
    [tau_peak(s), doppler_peak(s), S_max(s)] = complex_correlator(sats(s),data,t_vec,int_time,show_pl
    fprintf('    PRN %0.0f || Tau = %0.0f samples or %0.2f meters || Doppler = %0.0f Hz || S = %0.4f\
        sats(s),tau_peak(s),(tau_peak(s)/1023*0.001*3e8),doppler_peak(s),S_max(s));
end % s = 1:size(gps_ephem,1)



%% =========================================================================
% Problem 5 - Increase the integration time
%==========================================================================
int_time = 0.002; % 1ms
t_vec = 0 : tstep_sam : int_time;

sats = [5,6,12];
fprintf('Peaks with Integration Time 0.002 seconds:\n');
for s = 1:length(sats) % Look for all satellites
    [tau_peak_2(s), doppler_peak_2(s), S_max_2(s)] = complex_correlator(sats(s),data,t_vec,int_time,s
    fprintf('    PRN %0.0f || Tau = %0.0f samples or %0.2f meters || Doppler = %0.0f Hz || S = %0.4f\
        sats(s),tau_peak_2(s),(tau_peak_2(s)/1023*0.001*3e8),doppler_peak_2(s),S_max_2(s));
end % s = 1:size(gps_ephem,1)
```

```
function [delay, doppler, S_max] = complex_correlator(PRN,data,t_vec,int_time,show_plot)

fIF = -60e3;

mult = int_time/0.001;

% Create a vector of PRN2 C/A code values
CA = generate_CA_code(PRN,int_time);

% Match C/A code to time vector
tstep = int_time/length(CA);
sig_CA = zeros(1,length(t_vec));
for n = 1:length(t_vec)
    tval = t_vec(n);
    partial_index = tval/tstep;
    index = floor(partial_index)+1;
    if index > length(CA)
        index = index - length(CA);
    end
    sig_CA(n) = CA(index);
end


delay_vec = 0:length(t_vec);

dstep = 1000/(int_time*1000);
doppler_vec = -5e3:dstep:5e3;

S = zeros(length(delay_vec),length(doppler_vec));
for i = 1:length(delay_vec)/mult
    parfor j = 1:length(doppler_vec)
        tau = delay_vec(i);
        fD = doppler_vec(j);
        carrier_phase = 2*pi*(fIF + fD)*t_vec;

        sumS=0;
        for k=1:length(t_vec)
            ind = round(k+tau);
            sumS = sumS + data(ind)*sig_CA(k)*exp(-(1i)*carrier_phase(k));
        end
        S(i,j) = norm(sumS);
    end % j = 1:length(doppler_vec)
end % i = 1:length(delay_vec)

[vec_max,idx] = max(S);
[S_max,doppler_idx] = max(vec_max);
delay_idx = idx(doppler_idx);
```

```
delay = delay_vec(delay_idx);
doppler = doppler_vec(doppler_idx);

peak_doppler = S(:,doppler_idx);
peak_delay = S(delay_idx,:);

if show_plot == true
    % Plot 3D mesh
    fig = figure; hold on; grid on; grid minor;
    title(sprintf('Complex Correlator of PRN %0.0f',PRN));
    ylabel('Delay [samples]');
    xlabel('Doppler Frequency [Hz]');
    xlim([doppler_vec(1),doppler_vec(end)]);
    ylim([delay_vec(1),delay_vec(end)/mult]);
    zlabel('Magnitude');
    surf(doppler_vec,delay_vec,S);
    view(35,40)
    saveas(fig,sprintf('ASEN5090_HW9_PRN%0.0f_CC_%0.0f.png',PRN,int_time*1000),'png');

    % Plot Peak Doppler Bin
    fig = figure; hold on; grid on; grid minor;
    title(sprintf('Peak Doppler Bin of PRN %0.0f',PRN));
    xlabel('Delay [samples]');
    ylabel('Complex Correlator Magnitude');
    xlim([delay_vec(1),delay_vec(end)/mult]);
    plot(delay_vec,peak_doppler);
    saveas(fig,sprintf('ASEN5090_HW9_PRN%0.0f_PeakDoppler_%0.0f.png',PRN,int_time*1000),'png');

    % Plot Peak Delay Bin
    fig = figure; hold on; grid on; grid minor;
    title(sprintf('Peak Delay Bin of PRN %0.0f',PRN));
    xlabel('Doppler [Hz]');
    ylabel('Complex Correlator Magnitude');
    plot(doppler_vec,peak_delay);
    saveas(fig,sprintf('ASEN5090_HW9_PRN%0.0f_PeakDelay_%0.0f.png',PRN,int_time*1000),'png');

end


end
```