Matt Bachmeier

Table of Contents

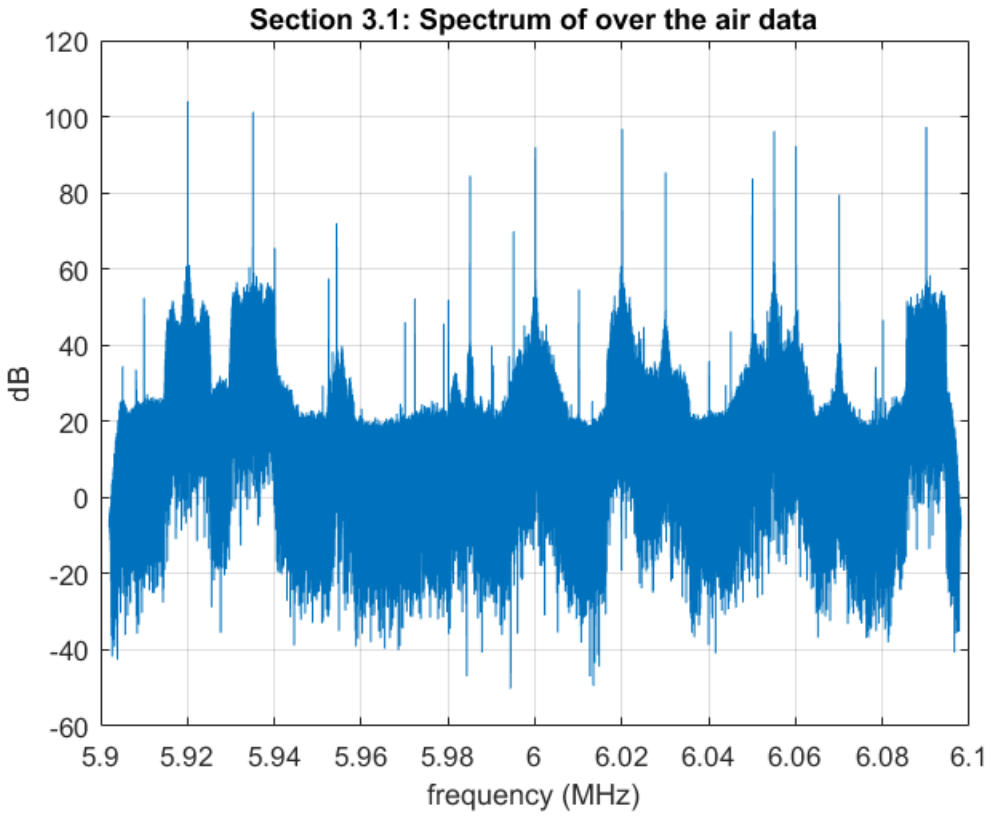
Lab 7 ECE 203 4/16/2017

Section 3.1

Plotting the spectrum

```
clear all
load shortwave.mat; % variable Fs, raw: N x 2
whos
Fs = 196078; % 196.078 kHz sampling frequency
f_LO = 6e6;  % carrier frequency
x = complex(raw(:,1),raw(:,2)); % complex baseband representation
% raw(:,1) is x_I and raw(:,2) is x_Q
X = fftshift(fft(x));
N = numel(X);
% write matlab code to plot abs(X) in db versus frequency in MHz
% The N points will be represented by a vector fvec corresponding to
% frequency range between f_LO - Fs/2 to f_LO + Fs/2 where
% f_LO = 6 MHz.
% Hint:  f/Fs = k/N

% <== Begin: enter your code here

fvec = linspace((f_LO - (Fs/2)), (f_LO + (Fs/2)), length(X));

% <== End: enter your code here

XdB = 20*log10(abs(X));
figure(1)
plot(fvec/1e6,XdB); grid on
title('Section 3.1: Spectrum of over the air data')
xlabel('frequency (MHz)')
ylabel('dB')
axis([5.9 6.1 -60 120])
```

```
  Name            Size                Bytes  Class     Attributes

  Fs              1x1                     8  double
  raw       4194304x2              67108864  double
```



Section 3.2

Type written the answers of questions 3.1 to 3.5 on a separate sheet. Submit these answers together with the published Matlab code.

Section 4.1

There is nothing to turn in for this sub-section.

Section 4.2

Estimating true carrier frequency f_1

```
% Note that the true carrier frquency has a large peak in the spectrum
 plot
% you have plotted in section 3.1. We will find that peak that is
 nearest
% to 0. The location of the peak gives |f_1 - f_LO|

% load shortwave.mat  % statement already executed in section 3.1
% f_LO = 6000e3;      % statement already executed in section 3.1
% x = raw(:,1) + 1i*raw(:,2); % statement already executed in section
 3.1
```

```
% X = fftshift(fft(x)); % statement already executed in section 3.1
% N = length(X); % statement already executed in section 3.1
% the following line from the lab handout is incorrect.
% freqs = linspace(f_LO - Fs/2,f_LO + Fs/2,N);
freqs = fvec; % we should just use the section 3.1 result.
% Now examine figure 1 and find a suitable frequency range say 5.9995
 to
% 6.0005 MHz within which, there is only one peak (max) value of Xdb.
% translate the indices (between 0:N) of freqs

% (Hint: find the indices of frequencies in frequency vector freqs
 that are
% closest to 5.9995 MHz = 5999500 Hz (minInd)
% and 6.0005 MHz = 6000500 Hz (maxInd)

% <== Begin: enter your code here

minInd = min(find(freqs <= 5999900 & freqs >= 5999500));
maxInd = max(find(freqs >= 6000000 & freqs <= 6000500));

% <== End: enter your code here

[~, ind0] = max(XdB(minInd:maxInd));
f_i = freqs(minInd+ind0-1);
fprintf('Estimated frequency is %g MHz\n',f_i/1e6)
```

```
Estimated frequency is 6.00007 MHz
```

Section 4.3 Implementing the demodulator

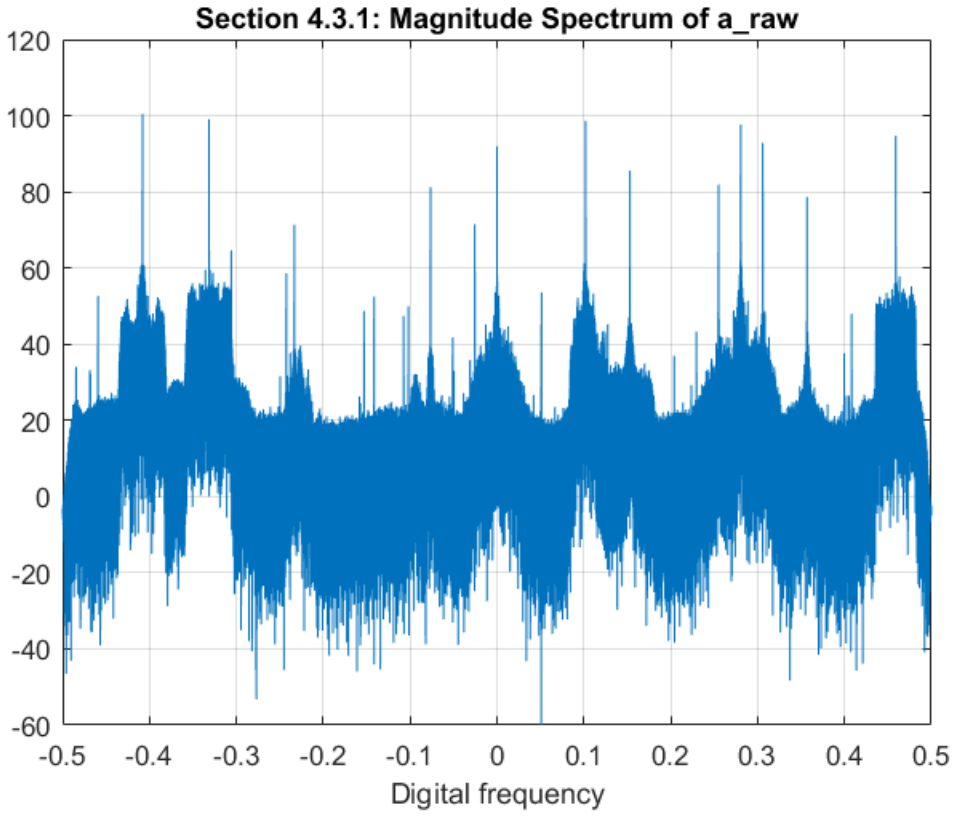
Section 4.3.1 Removing the Carrier Frequency f_i

referring to discussion in section 4.1, here we will implement step 1 and step 2

```
% Step 1. As described in eq. (4)
a_raw = x.*exp(-1i*2*pi*[1:N]'*(f_i - f_LO)/Fs); % from the handout
% plot the spectrum of a_raw (in dB unit).
% This is similar to the plot of spectrum of x
% except that the frequency axis is now digital frequency axis in
% [-0.5, 0.5-1/N]. You can use linspace to generate it.
% the x-axis digital frequency: dfreqs, spectrum of a_raw: AdB
figure(2),

% <== Begin: enter your code here

dfreqs = linspace(-1/2, 1/2, N);
AdB = 20*log10(abs(fftshift(fft(a_raw))));

% <== End: enter your code here
```

```
plot(dfreqs,AdB); grid on
xlabel('Digital frequency')
title('Section 4.3.1: Magnitude Spectrum of a\_raw')
axis([-0.5 0.5 -60 120])

% Now zoom in to check the peak is at 0. We simply repeat the section
 of code in
% section 4.2 with XdB replaced by AdB, f_i replaced by f_inew and
 ind0 by
% ind1

% <== Begin: enter your code here

[~, ind1] = max(AdB(minInd:maxInd));
f_inew = freqs(minInd+ind1-1);
fprintf('Estimated frequency is %g MHz\n',f_inew/1e6)

% <== End: enter your code here
```

Estimated frequency is 6 MHz



Section 4.3.2 Remove the phase

```
% this is done  by taking the real part of a_raw as described in the
% handout
a = real(a_raw);
```

```
soundsc(a, Fs);
```

Published with MATLAB® R2016a