Table of Contents

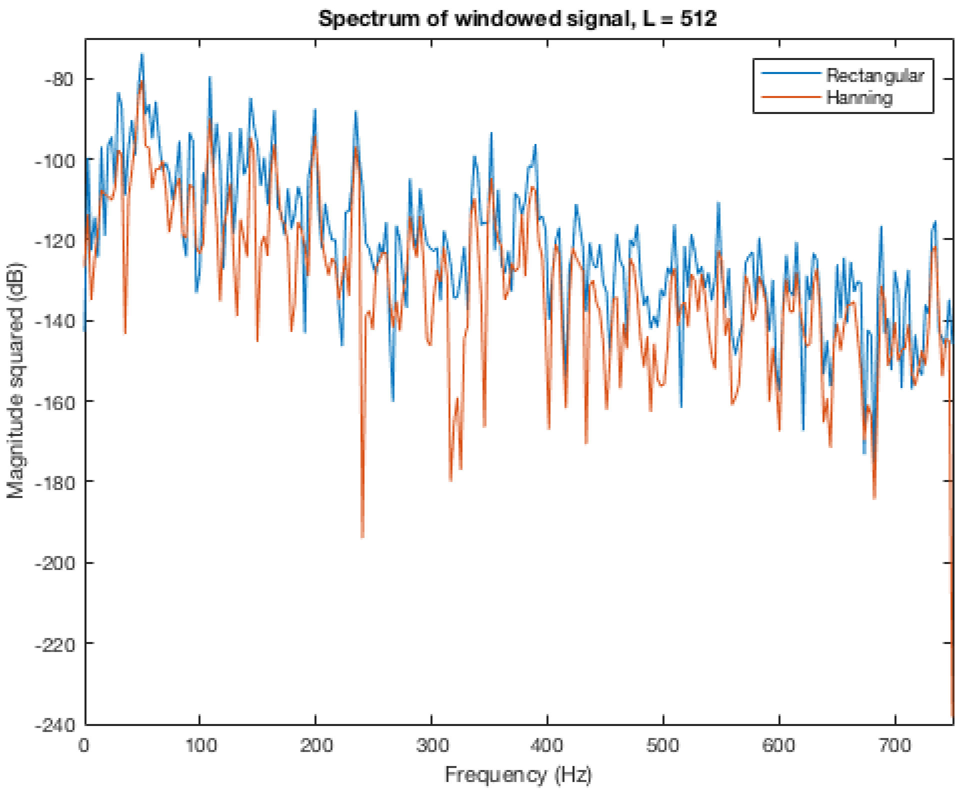

```
% Alex Christenson EE125: Digital Signal Processing Matlab Project 5
% Performing spectral analysis on measured data from an ocean
 acoustics
% experiment
```

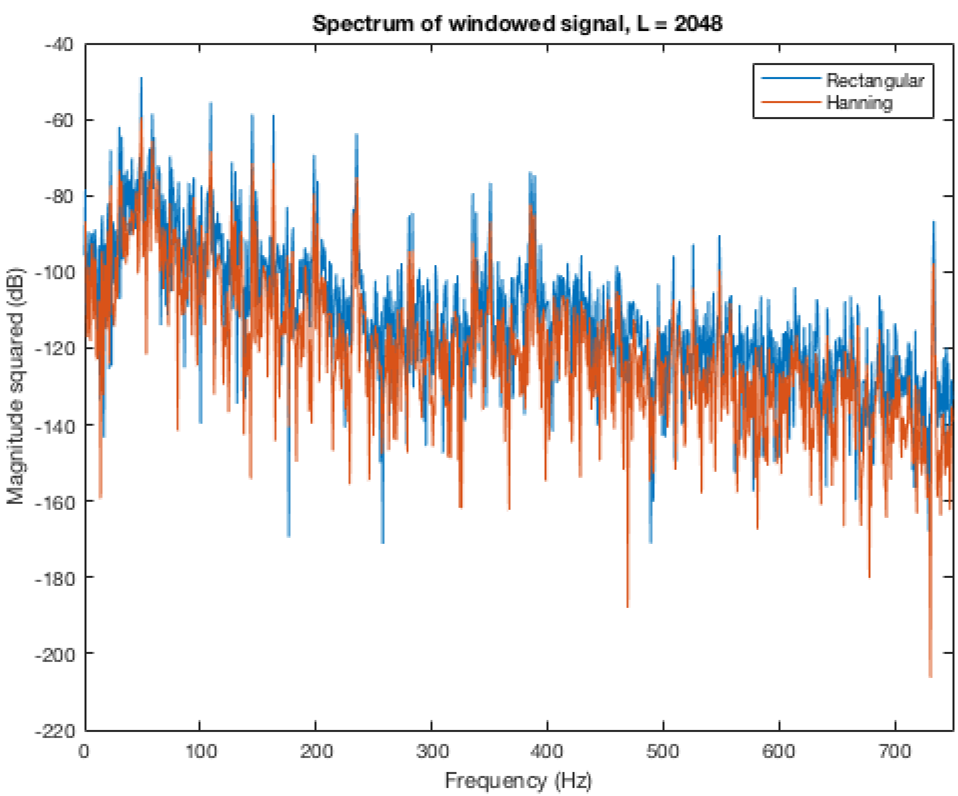
Part 1: DFT based spectral analysis

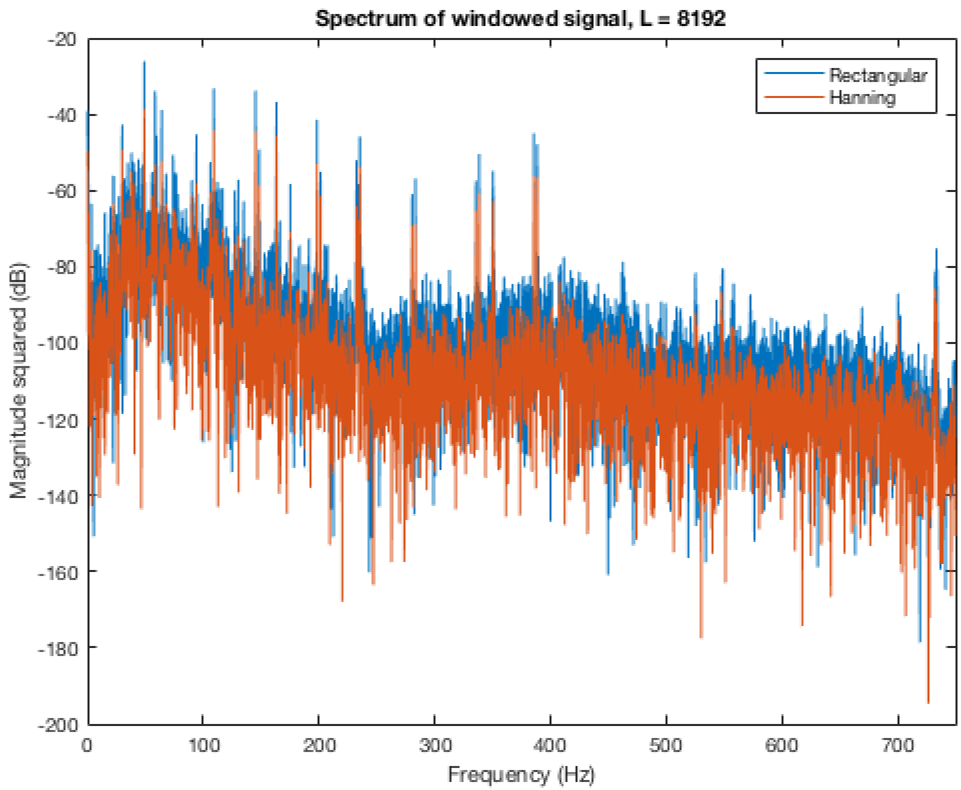
```
% Loading data from experiment
% Contains sample frequency (1.5 kHz) and xstart, which is the time
 series
% data for the start of the event, and xlater the time series data for
% after the event
load('Event59Data.mat')
x = xstart;

% Use FFT to perform spectral analysis for the signal
% Using a rectangular window
L = 512;
xRect = x(1:L).*rectwin(L);
XRect = fft(xRect,L);

% Using a hanning window
xHann = x(1:L).*hann(L);
XHann = fft(xHann,L);

% Plot the magnitude squared in dB of the two spectrums
f = Fs/2*linspace(0,1,L/2+1); % Create the frequency vector
XRectdB = mag2db(abs(XRect).^2); % Calculate mag squared and convert
 to dB
XHanndB = mag2db(abs(XHann).^2);

figure(1)

plot(f,XRectdB(1:L/2+1),f,XHanndB(1:L/2+1));

title('Spectrum of windowed signal, L = 512')
xlabel('Frequency (Hz)')
ylabel('Magnitude squared (dB)')
legend('Rectangular','Hanning')
axis([0 750 -240 -70])

% No, for the 512 window length you cannot clearly see that there are
 two
```

```
% tones around 200 Hz (at 198 and 201 Hz). For a window length of 512,
 the
% mainlobe width is (for rectangular window):
% 4pi/L = 4pi/512
% to convert from radians to Hz:
% 4pi/512 * Fs/2pi = 2*Fs/512 = 3000/512 = 5.8594 Hz.
% For the hanning window, mainlobe width is 8pi/M, which is twice that
 for
% the rectangular window, so the width is twice that for the
% rectangular window: 11.72 Hz.
% This tells us the bin width, and from this we can see that because
 the
% bin width is greater than the separation between the two
 frequencies,
% they will be unviewable.

% Repeating for L = 2048
L = 2048;
xRect2 = x(1:L).*rectwin(L);
XRect2 = fft(xRect2,L);

% Using a hanning window
xHann2 = x(1:L).*hann(L);
XHann2 = fft(xHann2,L);

% Plot the magnitude squared in dB of the two spectrums
f2 = Fs/2*linspace(0,1,L/2+1); % Create the frequency vector
XRectdB2 = mag2db(abs(XRect2).^2); % Calculate mag squared and convert
 to dB
XHanndB2 = mag2db(abs(XHann2).^2);

figure(2)

plot(f2,XRectdB2(1:L/2+1),f2,XHanndB2(1:L/2+1));

title('Spectrum of windowed signal, L = 2048')
xlabel('Frequency (Hz)')
ylabel('Magnitude squared (dB)')
legend('Rectangular','Hanning')
axis([0 750 -220 -40])

% Now that the window length is 2048 you can see the different toned
% frequencies. This makes sense because following the equation from
 the
% length equal to 512:
% 3000/2048 = 1.46 Hz, and for the Hann window, 2.93 Hz. This means
 the
% signals will be separable.

% Repeating for L = 2048
L = 8192;
xRect3 = x(1:L).*rectwin(L);
XRect3 = fft(xRect3,L);
```

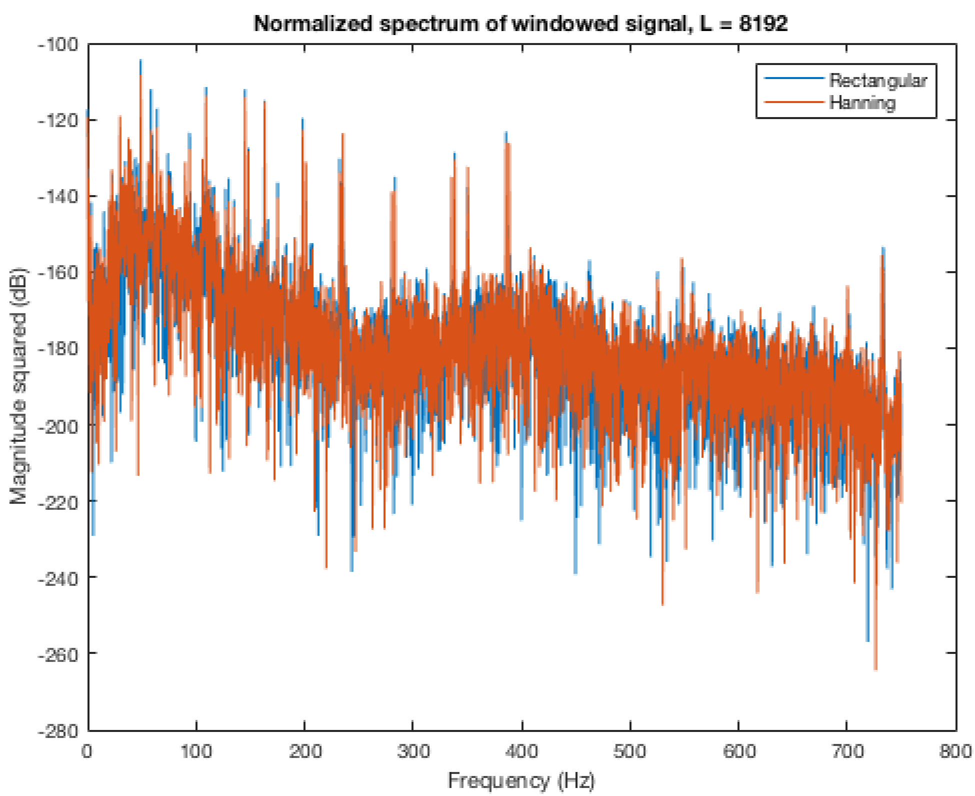
```
% Using a hanning window
xHann3 = x(1:L).*hann(L);
XHann3 = fft(xHann3,L);

% Plot the magnitude squared in dB of the two spectrums
f3 = Fs/2*linspace(0,1,L/2+1); % Create the frequency vector
XRectdB3 = mag2db(abs(XRect3).^2); % Calculate mag squared and convert
 to dB
XHanndB3 = mag2db(abs(XHann3).^2);

figure(3)

plot(f3,XRectdB3(1:L/2+1),f3,XHanndB3(1:L/2+1));

title('Spectrum of windowed signal, L = 8192')
xlabel('Frequency (Hz)')
ylabel('Magnitude squared (dB)')
legend('Rectangular','Hanning')
axis([0 750 -200 -20])

% Looking at the graph from further increasing the number of points in
 the
% window, the SNR does look like it improves because you can see more
% significant spikes from where the tones we want to pick out are, so
 it
% looks like the SNR is improving.

% Using the normalization factors
% Also, I used a different normalization factor
normRect = sum(rectwin(L).^2);
normHann = sum(hann(L).^2);
XRectdB3 = mag2db(abs(XRect3).^2/normRect); % Calculate mag squared
 and convert to dB
XHanndB3 = mag2db(abs(XHann3).^2/normHann);

figure(4)

plot(f3,XRectdB3(1:L/2+1),f3,XHanndB3(1:L/2+1));

title('Normalized spectrum of windowed signal, L = 8192')
xlabel('Frequency (Hz)')
ylabel('Magnitude squared (dB)')
legend('Rectangular','Hanning')
% axis([0 750 -200 -20])

% Yay they match more closely now :)
```


Spectrum of windowed signal, L = 512
Rectangular
Hanning
Magnitude squared (dB)
Frequency (Hz)
-80
-100
-120
-140
-160
-180
-200
-220
-240
0
100
200
300
400
500
600
700


Spectrum of windowed signal, L = 2048
Rectangular
Hanning
Magnitude squared (dB)
Frequency (Hz)
-40
-60
-80
-100
-120
-140
-160
-180
-200
-220
0
100
200
300
400
500
600
700



Spectrum of windowed signal, L = 8192
Rectangular
Hanning
Magnitude squared (dB)
Frequency (Hz)



Normalized spectrum of windowed signal, L = 8192
Rectangular
Hanning
Magnitude squared (dB)
Frequency (Hz)

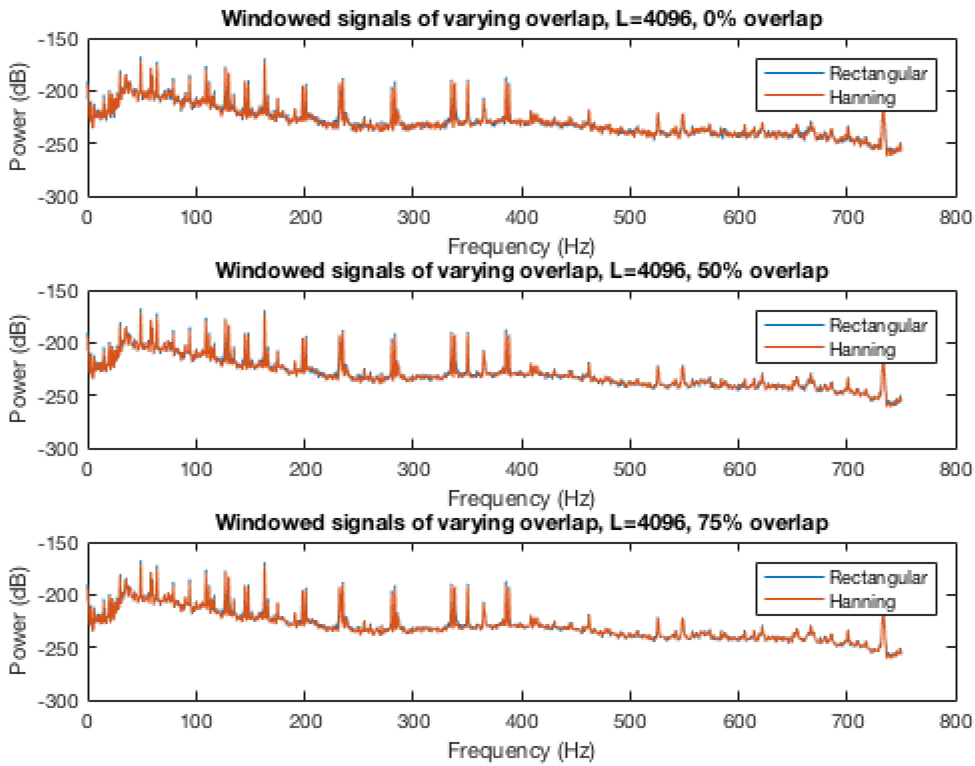
Part 2: Periodogram-based spectral analysis

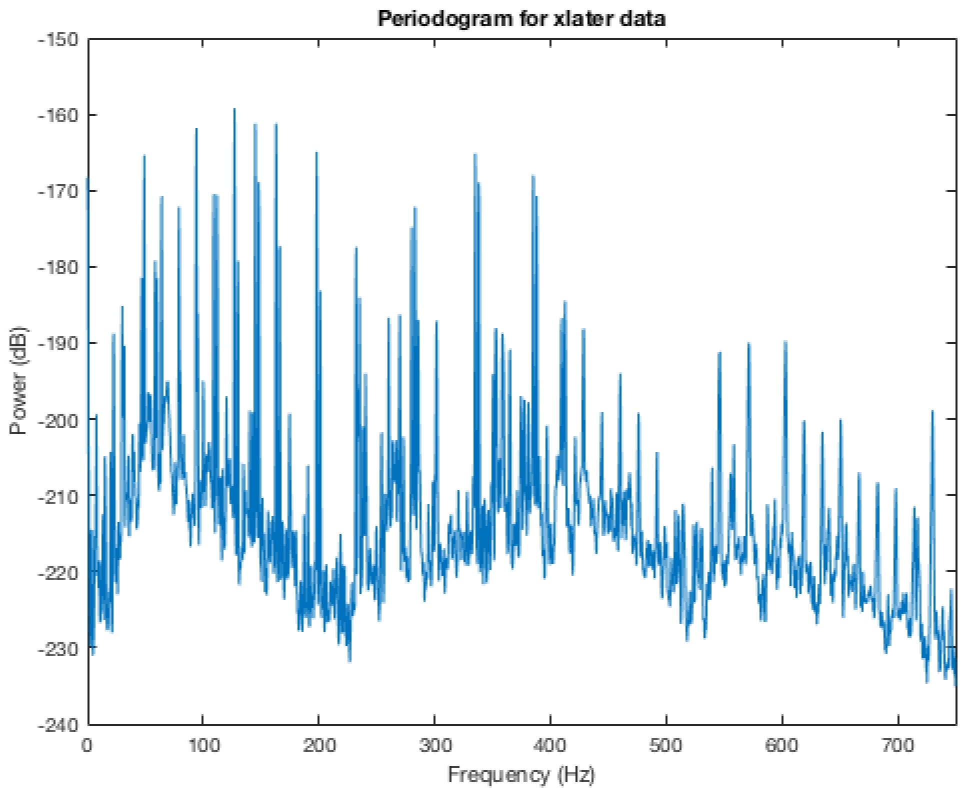
```
L = 4096;
[bartRect,f1] = pwelch(x,rectwin(L),0,L,Fs);
[bartHann,f2] = pwelch(x,hann(L),0,L,Fs);

[rect50,f1] = pwelch(x,rectwin(L),0.5*L,L,Fs);
[hann50,f2] = pwelch(x,hann(L),0.5*L,L,Fs);

[rect75,f1] = pwelch(x,rectwin(L),0.75*L,L,Fs);
[hann75,f2] = pwelch(x,hann(L),0.75*L,L,Fs);

figure(5)
subplot(3,1,1)
plot(f1,mag2db(bartRect),f2,mag2db(bartHann));
title('Windowed signals of varying overlap, L=4096, 0% overlap')
xlabel('Frequency (Hz)')
ylabel('Power (dB)')
legend('Rectangular','Hanning')

subplot(3,1,2)
plot(f1,mag2db(rect50),f2,mag2db(hann50));
title('Windowed signals of varying overlap, L=4096, 50% overlap')
xlabel('Frequency (Hz)')
ylabel('Power (dB)')
legend('Rectangular','Hanning')

subplot(3,1,3)
plot(f1,mag2db(rect75),f2,mag2db(hann75));
title('Windowed signals of varying overlap, L=4096, 75% overlap')
xlabel('Frequency (Hz)')
ylabel('Power (dB)')
legend('Rectangular','Hanning')

% It's hard to see from the graph (also apologies for not having a
 title
% for all of the subplots, but I'm using the new version of matlab and
% suptitle no longer works) but if you zoom in you can see the
 variance
% being reduced by the varying magnitudes in the noise around the
 signal
% becoming more consistent and approaching the expected value of the
 noise.
% As for the differences between the Hanning and Rectangular windows I
 know
% there probably should be some small difference but I can't see it
 really
% compound when looking at the graph.

% Will calculate quality factor and variance later
% Finding the number of windows for each percent overlap. I truncated
 and
% rounded down
```

```
K50 = floor(2*length(x)/L)-1;
K75 = floor(4*length(x)/L)-3;

% Storing the constants for overlap from harris paper
cRect50 = 50;
cRect75 = 75;
cHann50 = 16.7;
cHann75 = 65.9;

% Solving for eqn 19 in the paper
FHann50 = (1+2*cHann50^2)/K50 - 2*(cHann50^2)/K50^2;
FRect50 = (1+2*cRect50^2)/K50 - 2*(cRect50^2)/K50^2;
FHann75 = (1+2*cHann50^2+2*cHann75^2)/K75 - 2*(cHann75^2+2*cHann50^2)/
K75^2;
FRect75 = (1+2*cRect50^2+2*cRect75^2)/K75 - 2*(cRect75^2+2*cRect50^2)/
K75^2;

% Calculating the Q factors
QRect0 = floor(length(x)/L) % Signals are uncorrelated so just number
 of
% windows
QHann50 = 1/FHann50
QHann75 = 1/FHann75
QRect50 = 1/FRect50
QRect75 = 1/FRect75

% calculating the variance over an area of the graph with not many
 tonal
% signals
varHann50 = var(hann50(790:890))
varHann75 = var(hann75(790:890))
varRect50 = var(rect50(790:890))
varRect75 = var(rect75(790:890))

% If you compute the variance of the entire spectrum it doesn't really
 make
% sense because the intentional signals you're trying to communicate
 are
% going to skew your data depending on how much or what is being
% communicated more than the affect that noise has on the situation.

% Yeah I guess my variance factors are more or less consistent with
 the Q
% factors, but they are all really just lining up with the intuition I
 have
% about the situation, and yes the trends do line up well.

% Applying the periodogram to the xlater data
[pl,fl] = pwelch(xlater,hann(L),0.75*L,L,Fs);

figure(6)
plot(fl,mag2db(pl))
title('Periodogram for xlater data')
xlabel('Frequency (Hz)')
```

```
ylabel('Power (dB)')
axis([0 750 -240 -150])

% I did notice that there are a bunch of really powerful harmonics
% appearing in the spectral analysis now, and the variance for the
 noise
% might be much higher compared to the same averaging for the last
 data
% set.
```

```
QRect0 =

    24


QHann50 =

    0.0859


QHann75 =

    0.0103


QRect50 =

    0.0096


QRect75 =

    0.0059


varHann50 =

   1.8898e-25


varHann75 =

   1.7092e-25


varRect50 =

   2.2449e-25


varRect75 =

   2.0548e-25
```

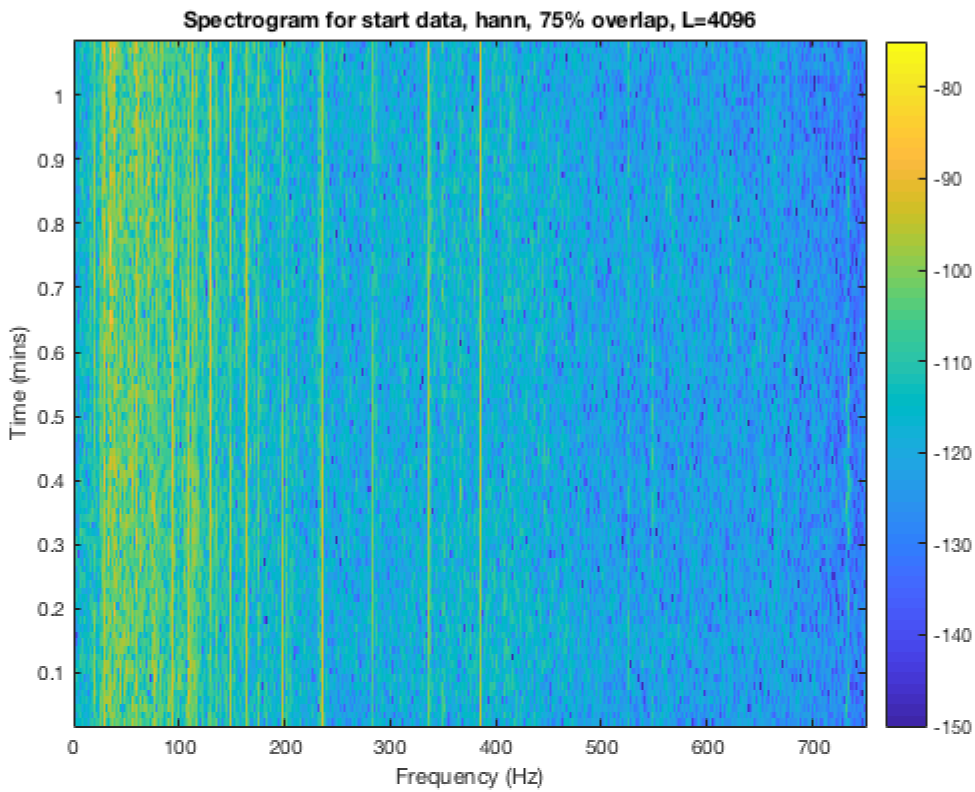

Windowed signals of varying overlap, L=4096, 0% overlap
Power (dB)
Frequency (Hz)
Rectangular
Hanning
Windowed signals of varying overlap, L=4096, 50% overlap
Power (dB)
Frequency (Hz)
Rectangular
Hanning
Windowed signals of varying overlap, L=4096, 75% overlap
Power (dB)
Frequency (Hz)
Rectangular
Hanning

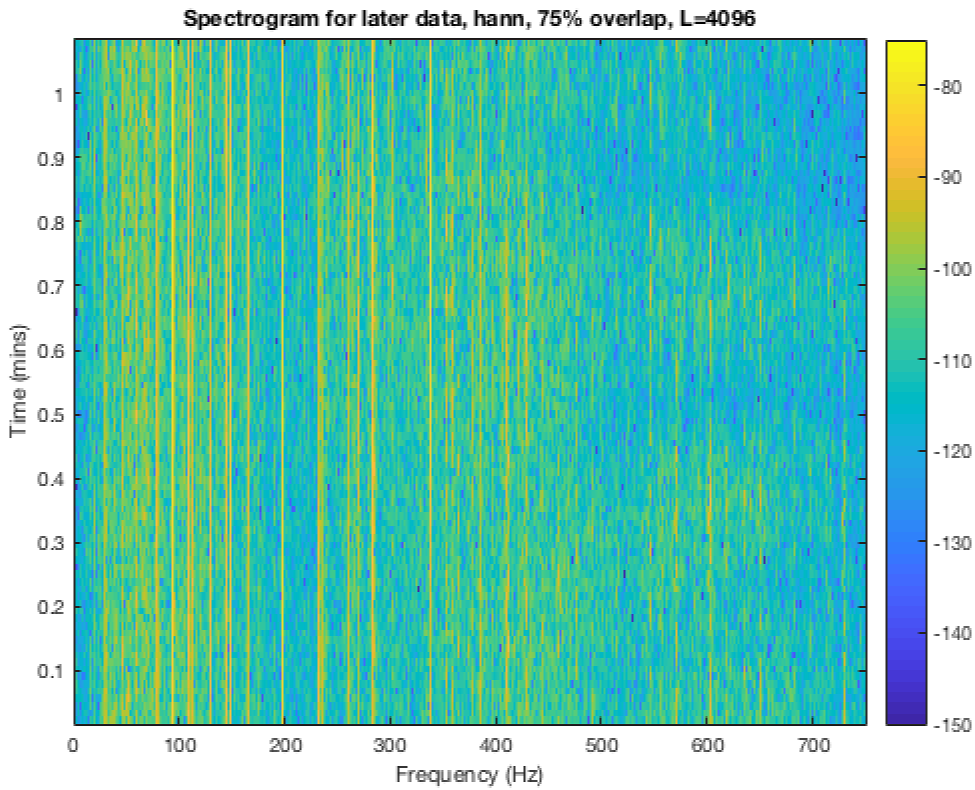


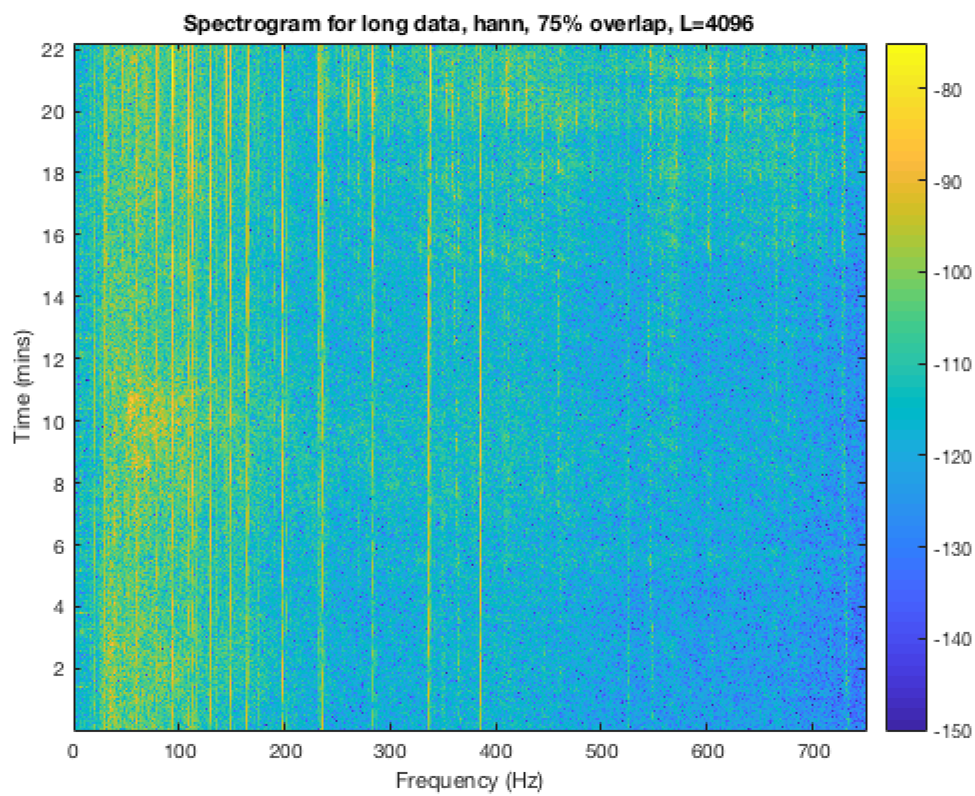
Part 3: Spectrogram

```
a = -150;
b = -75;

figure(7)
spectrogram(x,hann(L),0.75*L,L,Fs);
caxis([a b]); colorbar
title('Spectrogram for start data, hann, 75% overlap, L=4096')

figure(8)
spectrogram(xlater,hann(L),0.75*L,L,Fs);
title('Spectrogram for later data, hann, 75% overlap, L=4096')
caxis([a b]); colorbar

% The later data has a lot of consistency in the time domain and
 contains a
% lot more stronger components at regular intervals and higher
 frequencies.

load('Event59Data_20min.mat')
figure(9)
spectrogram(xbig,hann(L),0.75*L,L,Fs);
caxis([a b]); colorbar
title('Spectrogram for long data, hann, 75% overlap, L=4096')
```

```
% This spectrogram over a long period of data is really interesting
 because
% you can see where different events happen. Like some sort of event
% happens at around 10 minutes that causes a spike in a contained
 range of
% low frequency signals.
```




Spectrogram for later data, hann, 75% overlap, L=4096
Time (mins)
Frequency (Hz)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
100
200
300
400
500
600
700
-80
-90
-100
-110
-120
-130
-140
-150


Spectrogram for long data, hann, 75% overlap, L=4096
Time (mins)
Frequency (Hz)
22
20
18
16
14
12
10
8
6
4
2
0
100
200
300
400
500
600
700
-80
-90
-100
-110
-120
-130
-140
-150

Published with MATLAB® R2017b