Lab 3 - Dan Wortmann - February 17th, 2014

Table of Contents

Warm-Up: mychirp.m

```
%function [xx,tt] = mychirp( f1, f2, dur, fsamp )
%MYCHIRP generate a linear-FM chirp signal
%
% usage: xx = mychirp( f1, f2, dur, fsamp )
%
% f1 = starting frequency
% f2 = ending frequency
% dur = total time duration
% fsamp = sampling frequency (OPTIONAL: default is 11025)
%
% xx = (vector of) samples of the chirp signal
% tt = vector of time instants for t=0 to t=dur
%
if( nargin < 4 ) %-- Allow optional input argument
    fsamp = 11025;
end

tt = 0: 1/fsamp : dur;
%slope
mu = ((f2 - f1)/dur)/2;

psi = 2*pi*(f1*tt + (mu)*tt.*tt);
xx = real(exp(j*psi));

soundsc(xx)

%end
```

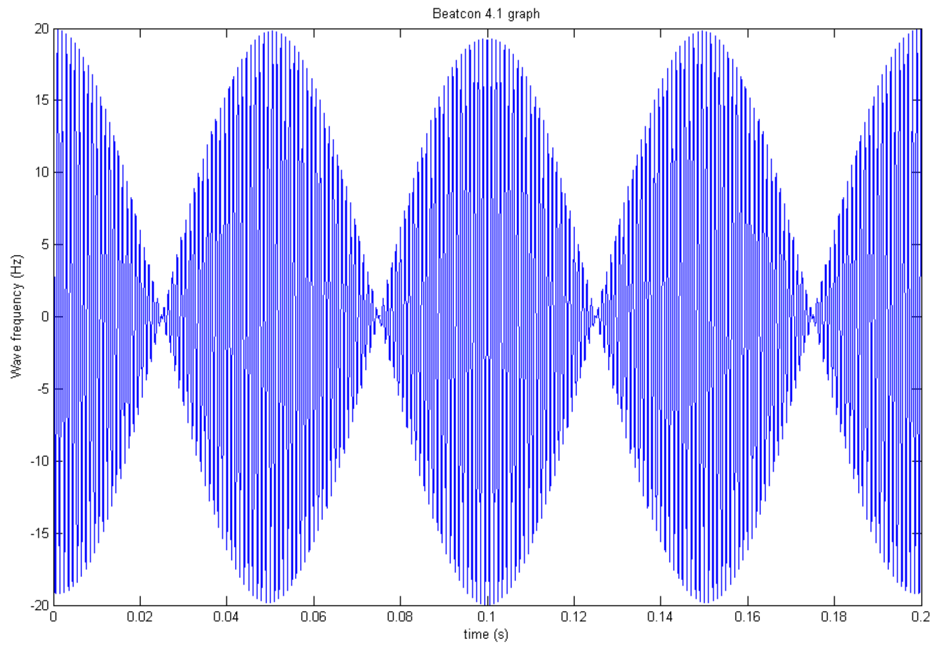
```
Undefined function or variable 'fsamp'.

Error in Lab3 (line 22)
tt = 0: 1/fsamp : dur;
```

4.1(a)

```
%function [xx, tt] = beat(A, B, fc, delf, fsamp, dur)
%BEAT compute samples of the sum of two cosine waves
% usage:
% [xx, tt] = beat(A, B, fc, delf, fsamp, dur)
%
% A = amplitude of lower frequency cosine
% B = amplitude of higher frequency cosine
% fc = center frequency
% delf = frequency difference
% fsamp = sampling rate
% dur = total time duration in seconds
% xx = output vector of samples
%--Second Output:
% tt = time vector corresponding to xx

%construct a vector of frequencies
fk = [fc - delf, fc + delf];

%construct a vector of complect amplitudes (no phase in this lab)
phi1 = 0;
phi2 = 0;
Xk = [A*exp(1j*phi1),B*exp(1j*phi2)];

% call the syn_sin function
[xx,tt] = syn_sin(fk, Xk, fsamp, dur);

%end
```

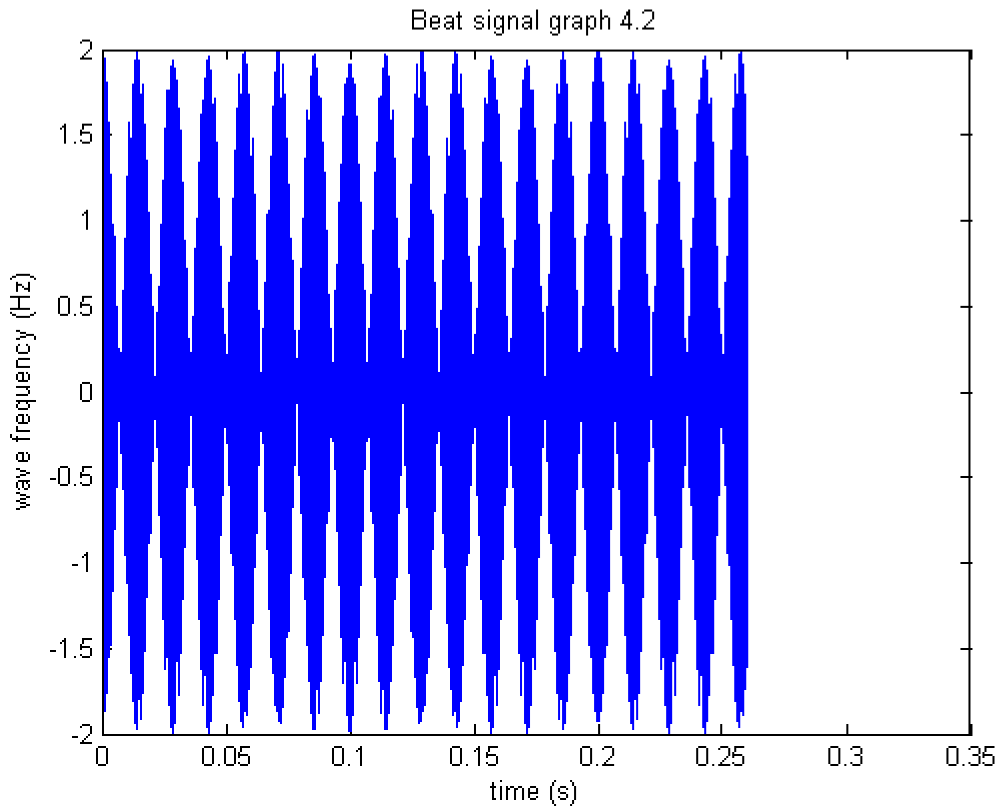
4.1(b)

```
[xx,tt] = beat(10,10,1000,10,11025,.2);

figure
plot(tt,xx)
title('Beatcon 4.1 graph');
xlabel('time (s)');
ylabel('wave frequency (Hz)');
```



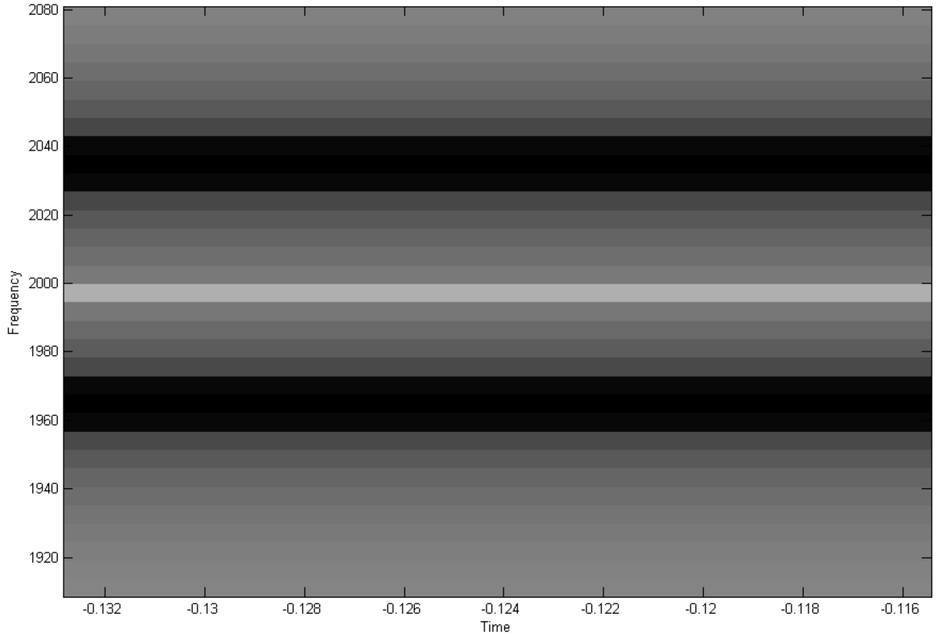
4.2(a)

```
[xx,tt] = beat(1,1,2000,35,11025,.26);

figure
plot(tt,xx)
title('Beat signal graph 4.2');
xlabel('time (s)');
ylabel('wave frequency (Hz)');
```



4.2(b)

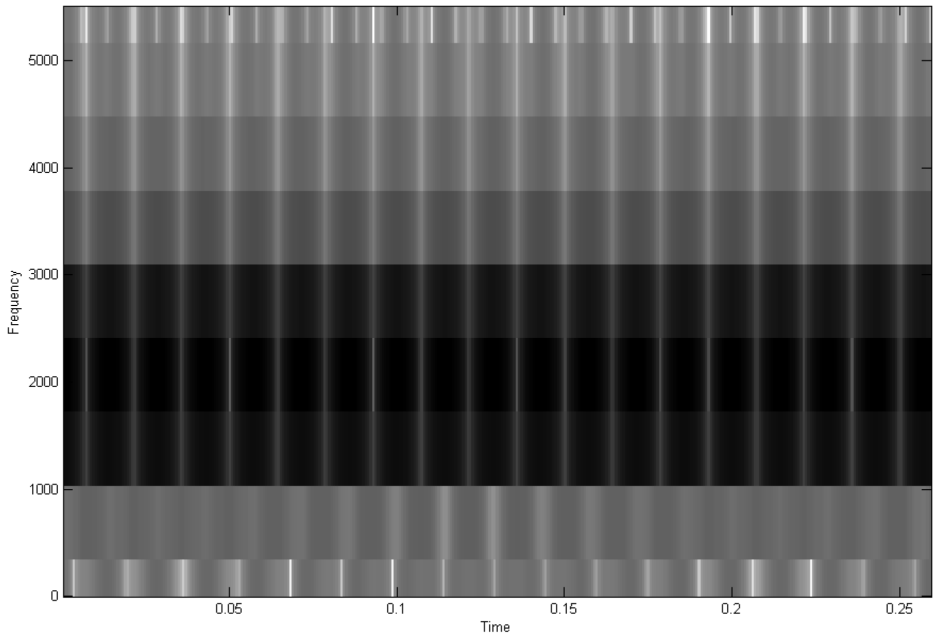
```
specgram(xx,2048,11025); colormap(1-gray(256))
```




```
%The spectrogram shows two definite clusters at the frequency around 2035
%and around 1965 Hz. This is validated by doing the computation for both
%the higher and lower frequency: fc +/- delf = 2032Hz and 1968Hz.
```

4.2(c)

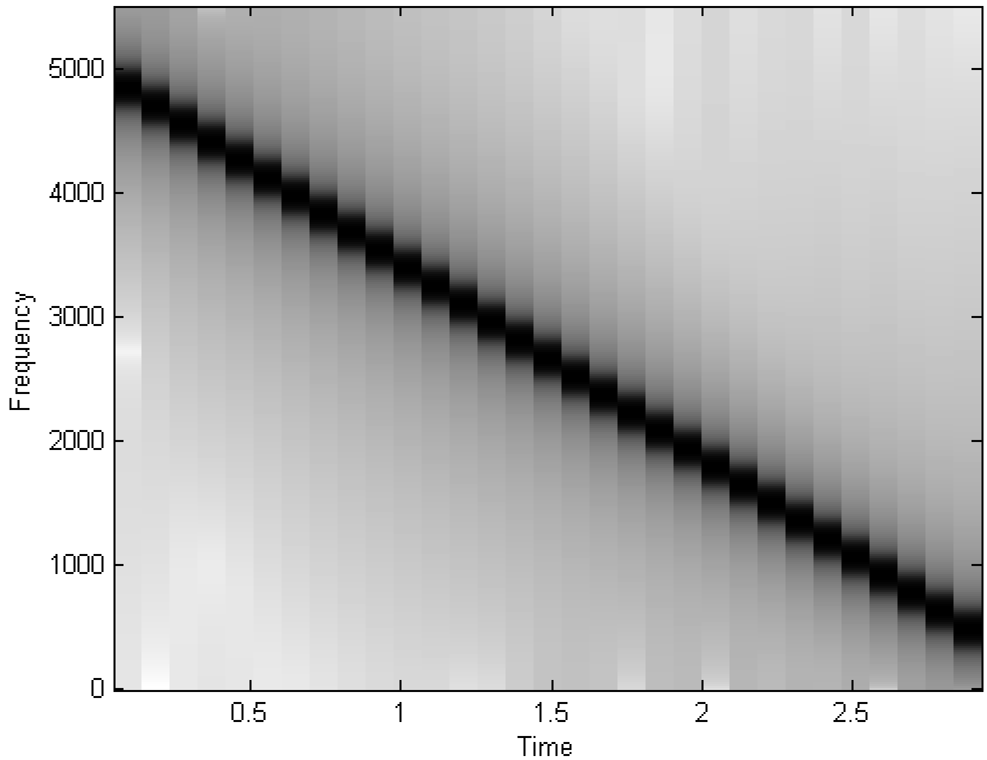
```
specgram(xx,16,11025); colormap(1-gray(256))
```




```
%This spectrogram is not nearly as accurate as the previous one. However,
%this still shows a definite cluster of frequencies around 1700 - 2300Hz.
%This is a very rough estimate.
```

4.3

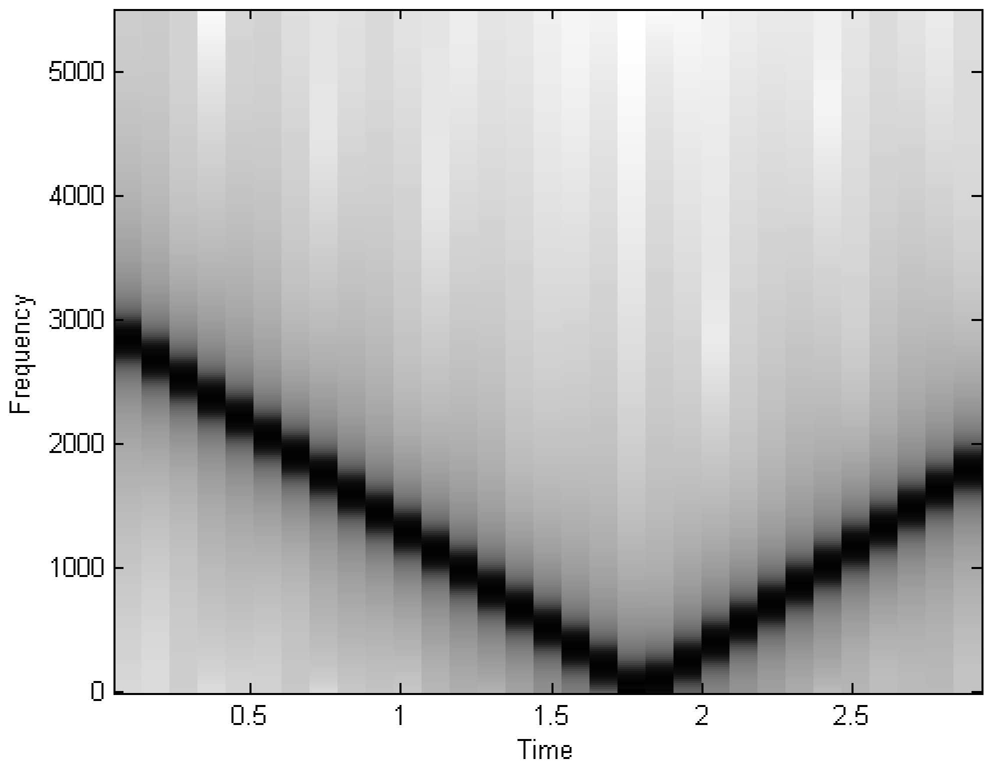
```
[xx,tt] = mychirp(5000,300,3,11025);

%The signal begins at a higher frequency (5000) and seems almost lounder.
%As time progresses the sound lowers in frequency and seems to get quieter
%to an extent. This change also happens linearly in time as the signal
%chirps 'down' to 300Hz.

specgram(xx,2048,11025); colormap(1-gray(256))
```




```
%The spectrogram reinforces my observations of the chirp. It not only shows
%the starting frequencies at 5000 and ending at 300, but also shows the
%linear change in frequency which can be deduced from the slop of the plot.
```

4.4

```
[xx,tt] = mychirp(3000,-2000,3,11025);
specgram(xx,2048,11025); colormap(1-gray(256))
```




```
%This time the chirp starts to go down from 3000Hz to 0, however as it
%reaches the low point it then begins to rise into a 2000Hz tone. This is
%accurate with the theory of the spectrum because in the spectrum the
%components of amplitude are taken for both the positive and negative values
%of a frequency. So despite the sound being at a -2000Hz, we are effectively
%listening to the absolute values of that frequency which is also 2000Hz.
```

Published with MATLAB® R2013a