
Matt Bachmeier

Table of Contents

Section 3.2 (Function of a Chirp)	1
Section 3.3 (Advanced Topic: Spectrogram)	1
Section 4.1(a) (Beat Notes)	2
Section 4.1(b) (Revised Instructions)	2
Section 4.2(a)	3
Section 4.2(b)	4
Section 4.2(c)	4
Section 4.3 (Spectrogram of a Chirp)	5
Section 4.4 (A chirp puzzle)	5

ECE 203 Lab 03: AM and FM Sinusoidal Signals 2/12/2017

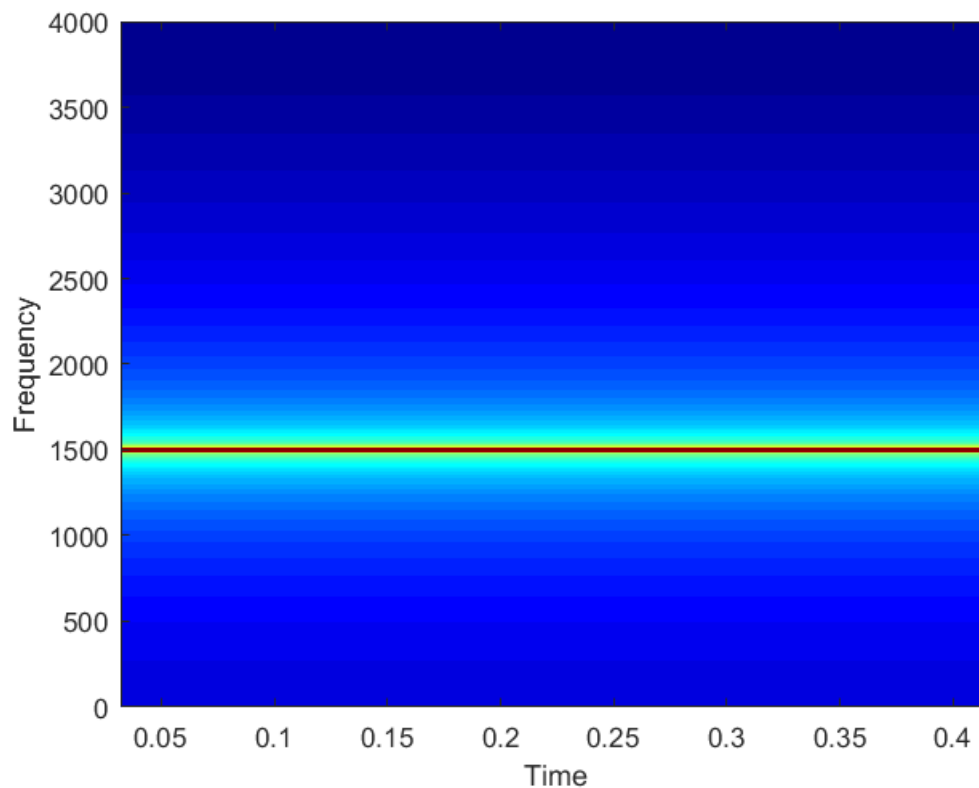
Section 3.2 (Function of a Chirp)

```
%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
%
%McClellan, Schafer, and Yoder, Signal Processing First, ISBN
  0-13-065562-7.
%Prentice Hall, Upper Saddle River, NJ 07458.
%c 2003 Pearson Education, Inc.3

%if( nargin < 4 ) %-- Allow optional input argument
%fsamp = 11025;
%end
%dt = 1/fsamp;
%tt = 0:dt:dur;
%slope = (f2 - f1)/dur;
%cosf = 2*pi*(100 + f1*tt + (slope/2)*tt.*tt);
%xx = real(1 * exp(j*cosf));
```

Section 3.3 (Advanced Topic: Spectrogram)

```
fs=8000; xx = cos(3000*pi*(0:1/fs:0.5)); spectrogram(xx,1024,fs);
```



Section 4.1(a) (Beat Notes)

```
%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

%tt = 0:1/fsamp:dur;
%xx = syn_sin([fc-delf, fc+delf], [A,B], fsamp, dur, 0);
%end
```

Section 4.1(b) (Revised Instructions)

```
A=10; B=10; fc=1000; delf=10; fsamp=11025; dur=1;
[xx,tt] = beat(A,B,fc,delf,fsamp,dur);
```

```
nvec = find(tt<=.2);
figure('Position', [290 220 890 300])
plot(tt(nvec), xx(nvec)); grid on
title(['Sum of cosines: x(t) = ' num2str(A) ' cos(2\pi ' num2str(fc-
delf) ' t) + ' num2str(B) ' cos(2\pi ' num2str(fc+delf) ' t)'])
xlabel('Time (s)')
ylabel('Amplitude')

% The period of the envelope based on the graph is 0.1 seconds
% because it is double the peak to peak value due to the fact that
% it is the overlay of two cosine signals. The period of the high
% frequency signal is 0.001 seconds based on the graph. This makes
% sense because the delta frequency is 10 Hz and the frequency of
% the high frequency signal is fc, hence 1000 Hz.

EnvelopePeriod = 1/delf
HighFrequencyPeriod = 1/fc

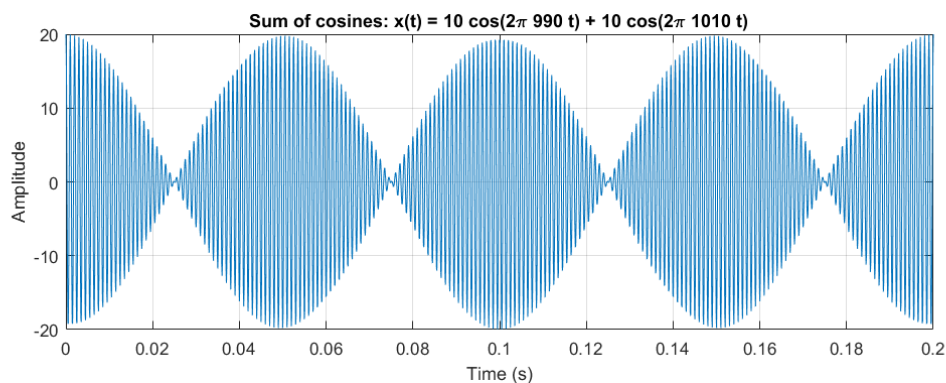
Warning: Imaginary parts of complex X and/or Y arguments ignored

EnvelopePeriod =

    0.1000

HighFrequencyPeriod =

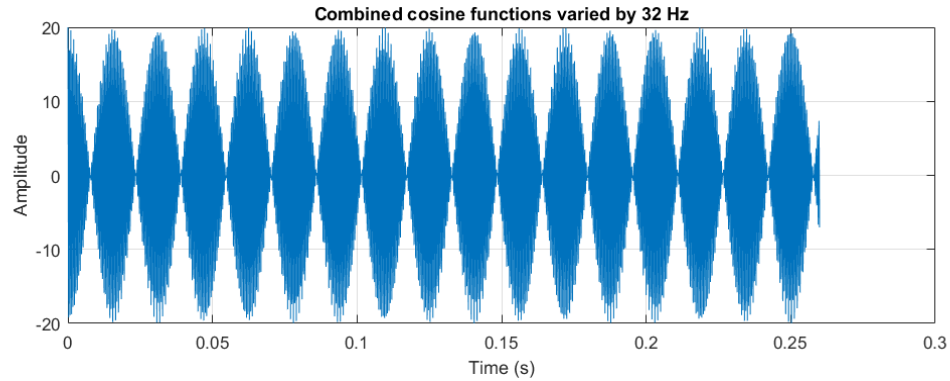
    1.0000e-03
```



Section 4.2(a)

```
A=10; B=10; delf=32; dur=0.26; fsamp=11025; fc =2000;
[x2,t2] = beat(A,B,fc,delf,fsamp,dur);
plot(t2,x2); grid on
title('Combined cosine functions varied by 32 Hz')
xlabel('Time (s)')
ylabel('Amplitude')

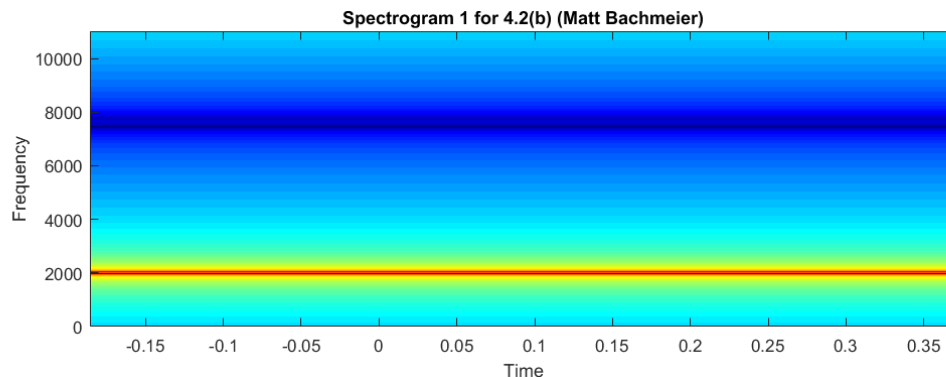
Warning: Imaginary parts of complex X and/or Y arguments ignored
```



Section 4.2(b)

```
specgram(x2,2048,fsamp);
title('Spectrogram 1 for 4.2(b) (Matt Bachmeier)');

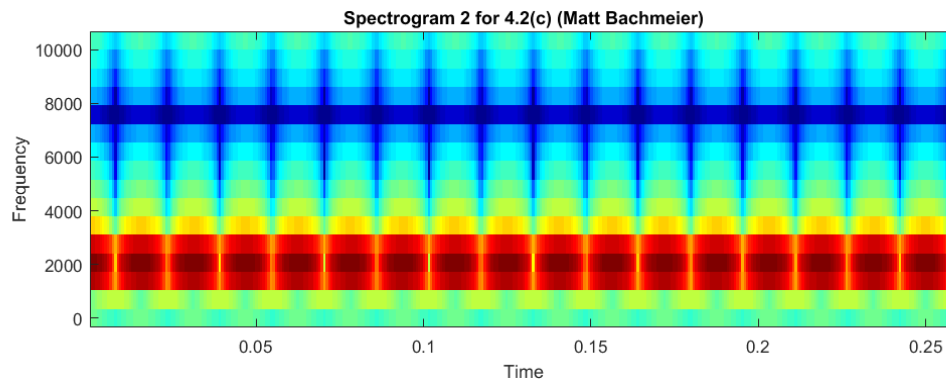
% The spectrogram shows the correct frequencies used.
% There is a solid line at around 2000 Hz and when you zoom
% in on it there are two distinct lines at 2032 and 1968, which
% makes sense because the center frequency was 2000 with a change of
% 32.
```



Section 4.2(c)

```
specgram(x2, 16, fsamp);
title('Spectrogram 2 for 4.2(c) (Matt Bachmeier)');

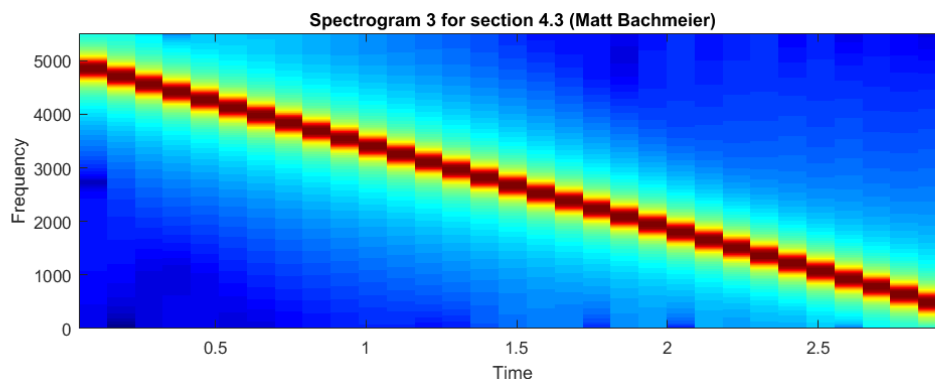
% This spectrogram is much less clear than the one previous.
% The one before had 2 distinct lines where the frequencies were,
% but this one looks messy with changing colors and lines throughout
% the
% whole thing. The larger window lets us see the two similar
% frequencies
% separately.
```



Section 4.3 (Spectrogram of a Chirp)

```
x3 = mychirp(5000, 300, 3, 11025);
soundsc(x3)
specgram(x3, 2048, 11025);
title('Spectrogram 3 for section 4.3 (Matt Bachmeier)');
```

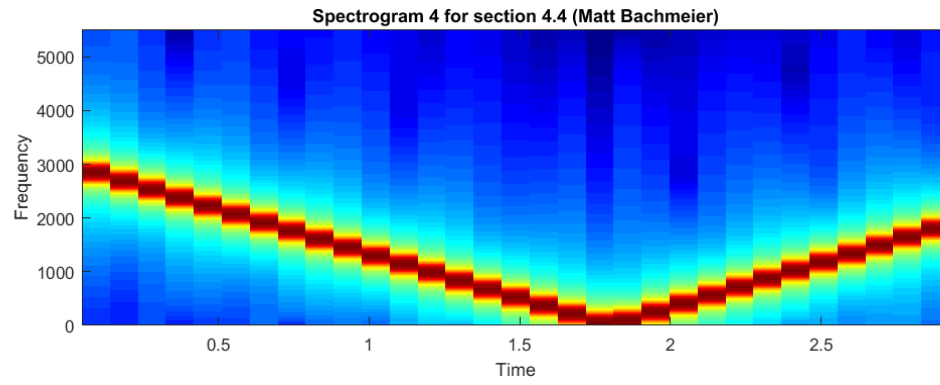
```
% The chirp decreases linearly. The spectrogram shows this
% through the linearly decreasing line passing through the
% spectrum from 5000 Hz to 300 Hz.
```



Section 4.4 (A chirp puzzle)

```
x4 = mychirp(3000, -2000, 3, 11025);
soundsc(x4)
specgram(x4, 2048, 11025);
title('Spectrogram 4 for section 4.4 (Matt Bachmeier)');
```

```
% When this chirp is played it decreases then increases again. This
% is shown by the spectrogram where it decreases from 3000 to 0,
% then back up to 2000, all linearly. This is explained by the fact
% that a negative frequency has a corresponding positive part with the
% complex conjugate.
```



Published with MATLAB® R2016a