

### Lab 03: Beat Notes and Time-Varying Signals

#### Introduction

For lab 03, we were introduced to more complicated signals that are related to the basic sinusoid. Some examples of these signals exist via radio and TV such as frequency modulation (FM) and amplitude modulation (AM). In addition, these signals can create interesting sounds that are similar to musical instrument notes.

#### Pre-lab

Beat note: multiply 2 sinusoids having different frequencies  
 it creates a beat note sounds like a warble.

Chirp signal: swept-frequency signal whose frequency changes linearly from some low value to a high one.

3.17  
 $\omega_1 = 2\pi f_1$  to  $\omega_2 = 2\pi f_2$  ; time from  $t = 0$  to  $t = T_2$

general formula:  $x(t) = A \cos(\alpha t^2 + \beta t + \phi) = \cos(\psi(t))$   
 for chirp  $\psi(t) = \alpha t^2 + \beta t - \phi$

derivative of  $\psi(t)$  is the instantaneous freq.  
 $\omega_i(t) = \frac{d}{dt} \psi(t)$  rad/sec

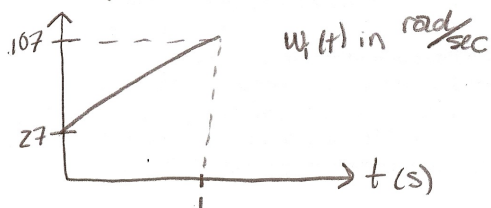
a) formulas for beginning freq. ( $\omega_1$ ):  $\omega_1 = \omega_i(t)|_{t=0} = \beta$   
 " ending freq. ( $\omega_2$ ):  $\omega_2 = \omega_i(t)|_{t=T_2} = 2\alpha T_2 + \beta$

$\omega_i(t) = \frac{d}{dt} \psi(t) = 2\alpha t + \beta$

b) chirp signal:  $x(t) = \text{Re}\{e^{j(40t^2 + 27t + 13)}\}$   
 derive formula for instantaneous freq. vs. time.

$\psi(t) = 40t^2 + 27t + 13$   
 $\omega_i(t) = \frac{d}{dt} \psi = 80t + 27$

c) plot of instantaneous freq. (in Hz) vs. time over range  $0 \leq t \leq 1$  s



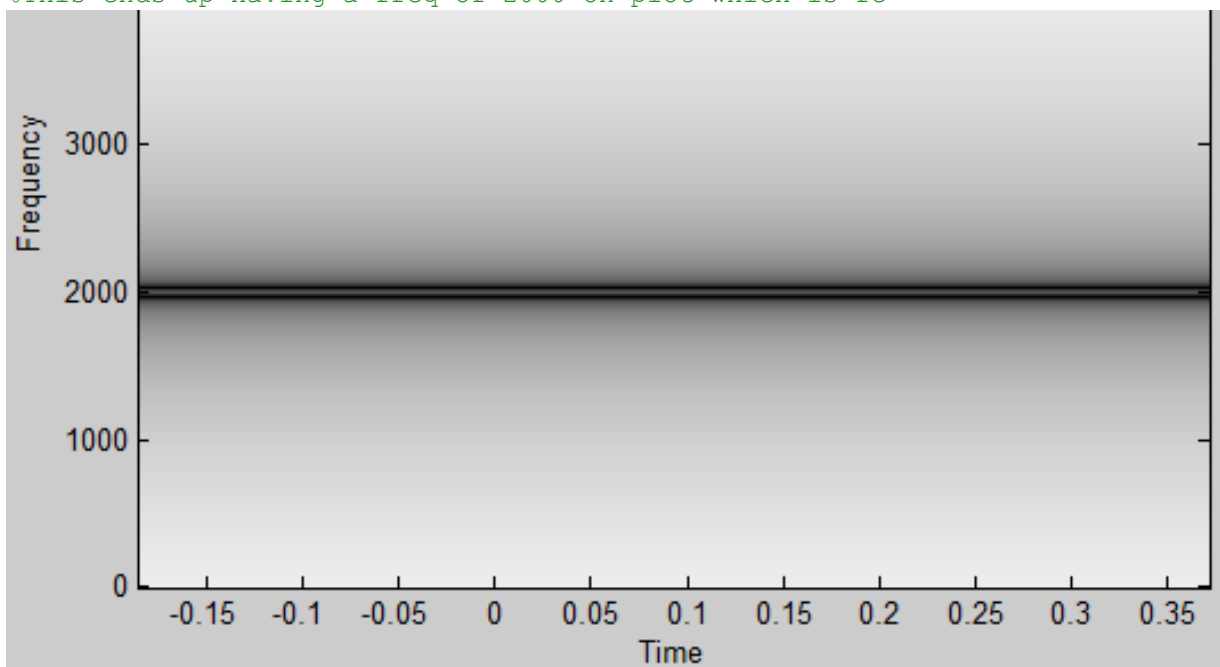
### Part 1: Beat Notes

Beat notes show one way to investigate the time-frequency characteristics of spectrograms. Some frequencies can be found presently and vary with time. A spectrogram estimates the frequency content over short sections of the signal. For example, if the section length is very short, we can track rapid changes in the frequency. Even shorter sections lack the ability to do accurate frequency measurement because the amount of input data is limited. Long sections can give great frequency measurements, but can't track frequency changes very well. During part 1, we used the equation for a beat signal

$$x(t) = A \cos(2\pi(f_c - f_\Delta)t) + B \cos(2\pi(f_c + f_\Delta)t)$$

to view the beat note as a single frequency sign whose amplitude varies with time or as two signals with different constant frequencies.

```
fsamp = 11025; %set sampling frequency
dt= 1/fsamp; % set sampling interval in seconds
dur=.26; %set signal duration in seconds
t= 0:dt:dur;
fd=32;
fc=2000;
A=1;
B=1;
x=A*cos(2*pi*(fc-fd)*t)+B*cos(2*pi*(fc+fd)*t);
plot(t,x,'-b');
grid on
soundsc(x,11025);
specgram(x,2048,11025);
colormap(1-gray(256));
%This ends up having a freq of 2000 on plot which is fc
```



As you can see above, there are two signals with constant frequencies around 2000 Hz.

## Part 2: Investigating Chirp Signals

This second part of the lab investigates chirp signals. A chirp signal is a sinusoid whose frequency changes linearly from a starting value to an ending one. This type of signal is used a lot in FM transmission.

### Part 2a: Investigating Chirp Signals

The total duration of the synthesized signal and the length of the tt vector can be found by using the following code in MATLAB:

```
fsamp = 11025; %set sampling frequency
dt= 1/fsamp; % set sampling interval in seconds
dur=1.5; %set signal duration in seconds
tt= 0:dt:dur; %create vector of time samples spaced at dt seconds
psi= 2*pi*(100+200*tt+500*tt.*tt); %set argument for chirp function
xx=7.7*cos(psi); %modulate signal with cosine and amp 7.7
soundsc(xx,fsamp); %play signal
specgram(xx,1024,fsamp); %plot spectrum
grid on
```

```
A=fsamp*dur %total duration
B=length(tt) %length of tt vector
```

#### Output:

```
A = 1.6538e+04
B = 16538
```

### Part 2b: Investigating Chirp Signals

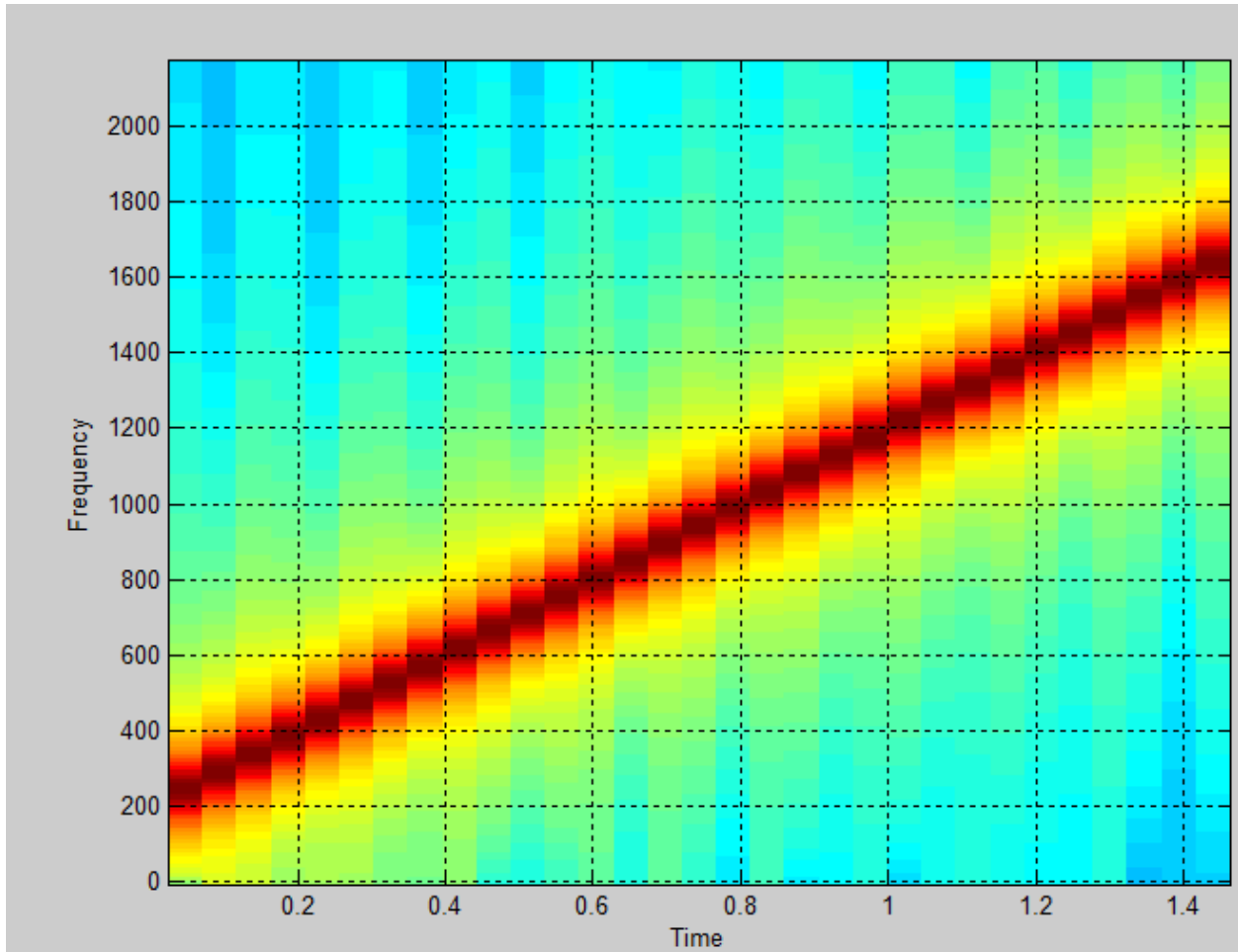
Signals can only be synthesized by finding the signal's defining formula at discrete instants of time in MATLAB. These are samples of the signal and we use the following formula

for a chirp:  $x(t_n) = A \cos(2\pi\mu t_n^2 + 2\pi f_0 t_n + \phi)$ . The values for:  $t_n = tt$ ,  $A = 7.7$ ,  $\mu = 500$ ,  $f_0 = 200$ ,  $\phi = 200\pi$ .

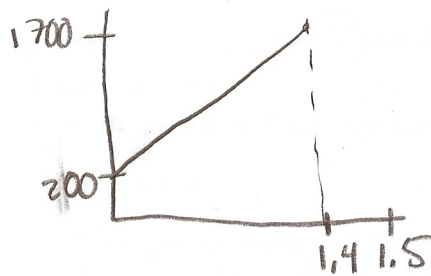
### Part 2c: Investigating Chirp Signals

For part c, I found the range of frequencies that will be synthesized by the script in part a. Below you can find a sketch of instantaneous frequency vs. time. The minimum frequency heard is 200 Hz and the maximum frequency heard is 1700 Hz.

```
fsamp = 11025; %set sampling frequency
dt= 1/fsamp; % set sampling interval in seconds
dur=1.5; %set signal duration in seconds
tt= 0:dt:dur; %create vector of time samples spaced at dt seconds
psi= 2*pi*(100+200*tt+500*tt.*tt); %set argument for chirp function
xx=7.7*cos(psi); %modulate signal with cosine and amp 7.7
soundsc(xx,fsamp); %play signal
specgram(xx,1024,fsamp); %plot spectrum
grid on
```



\*\*Correction: on this graph, Frequency is in Hz and time is in seconds.\*\* As you can see above, the frequency is linear starting at 200 and going to 1700 Hz. Below are the calculations explaining why:



$$x(t_n) = 7.7 \cos(2\pi(500t^2 + 200t + 100))$$

$\psi(t)$

$$\frac{d\psi}{dt} = 1000t + 200$$

so, if  $\begin{cases} t=0, \text{ Hz}=200 \\ t=1.5, \text{ Hz}=1700 \end{cases}$

### **Part 2d: Investigating Chirp Signals**

After listening to the chirp signal, I found the signal's frequency to be increasing. You can also prove that the signal's frequency is increasing by looking at the above graph.

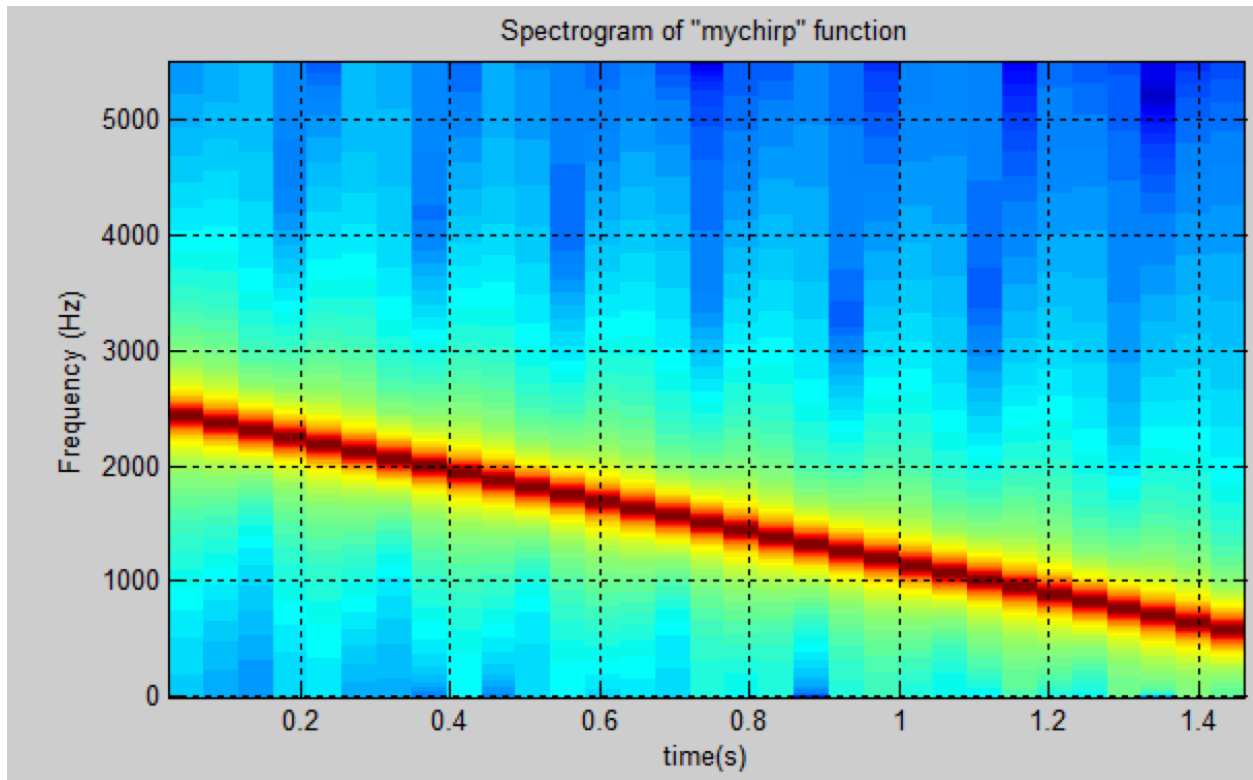
### **Part 2e: Investigating Chirp Signals**

For testing purposes, I formulated a chirp sound with a starting frequency at 2500 Hz and ending at 500 Hz lasting 1.5 seconds. The chirp sounded to be chirping down. The following code shows this:

```
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 ) %This allows optional input argument
    fsamp = 11025; %Set frequency
end
dt = 1/fsamp; %Sets the sampling interval in seconds
tt = 0:dt:dur; %Creates a vector of time samples with spacing of dt seconds
f0 = f1; %Finding the value that produces the starting frequency
yy = (f2 - f0)/(2*dur); %Find the value that produces the ending frequency
psi = (2*pi*f0*tt+2*pi*yy*tt.*tt); %Set argument for the chirp function
xx = 7.7*cos(psi); %Modulate signal with cosine and amplitude of 7.7
soundsc(xx, fsamp);
specgram(xx, 1024, fsamp), title('Spectrogram of "mychirp" function');
grid on
xlabel('time(s)');
ylabel('Frequency (Hz)');

end

mychirp(2500,500,1.5,11025);
```



### Part 2f: Investigating Chirp Signals

Using the function code formed above, I again synthesized a chirp signal lasting 3 seconds with a sampling frequency of 11025 Hz. The signal starts at a frequency of 5000 Hz and ending at 300 Hz. After listening to the chirp, it sounds like the frequency movement is linearly down. A spectrogram of this signal can be found below:

```
mychirp(5000,300,3,11025);
```

