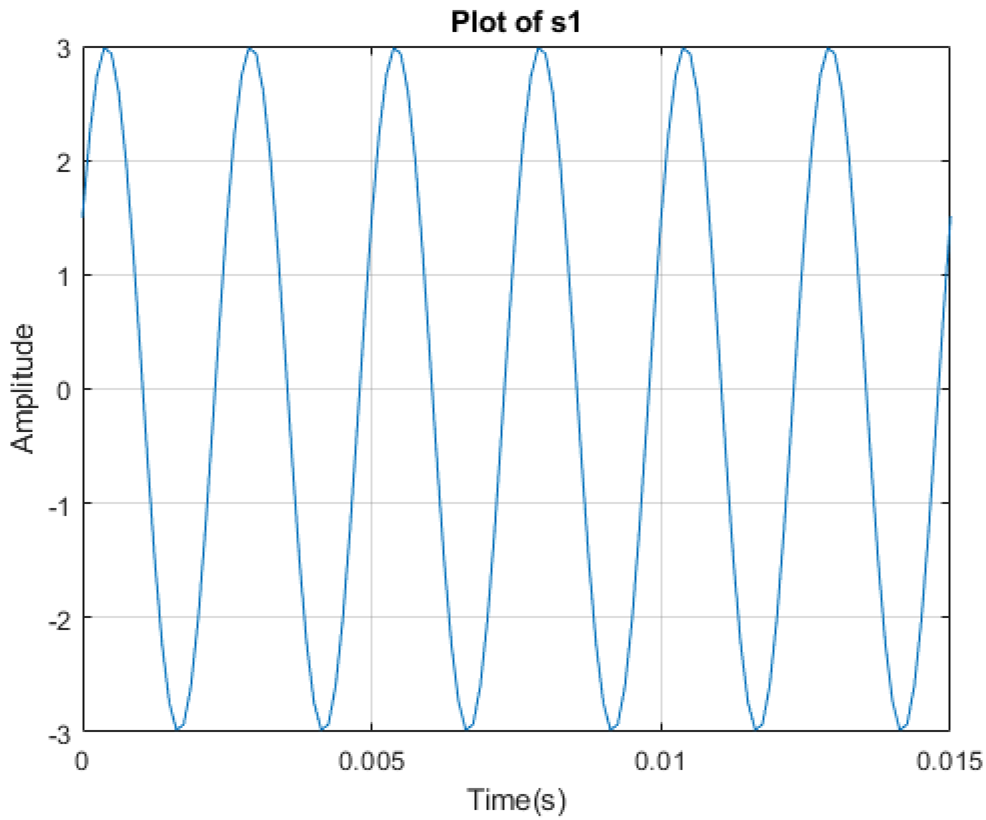
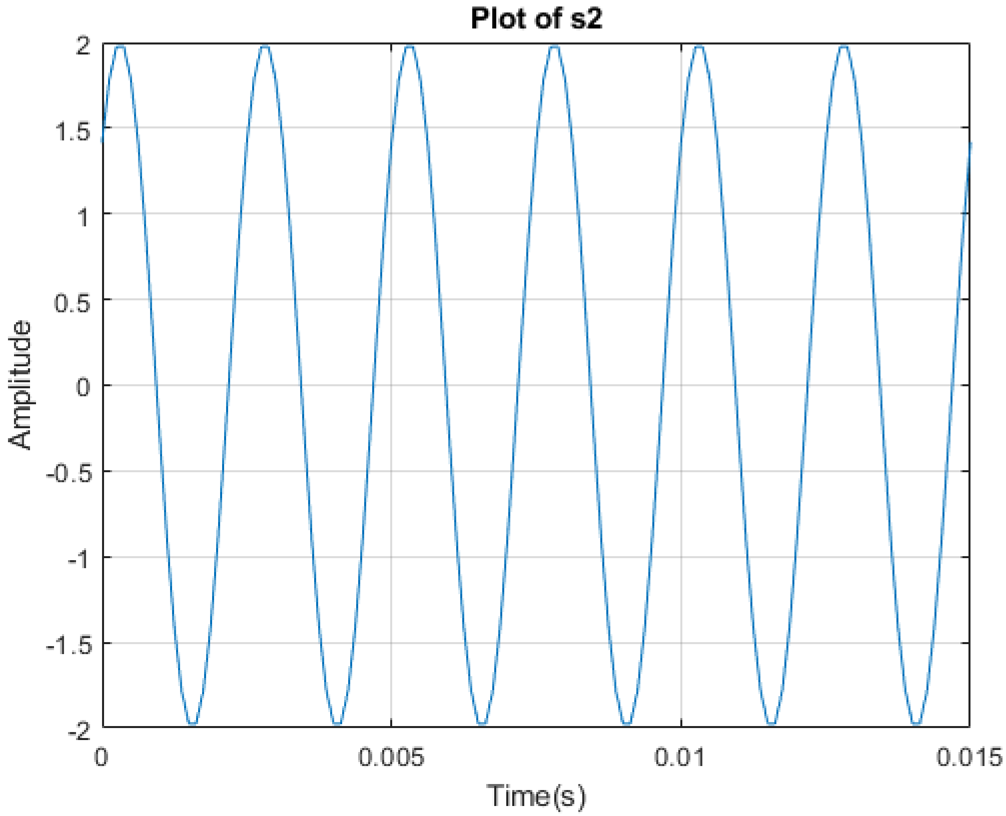
Exercise 2.1 Koby Miller

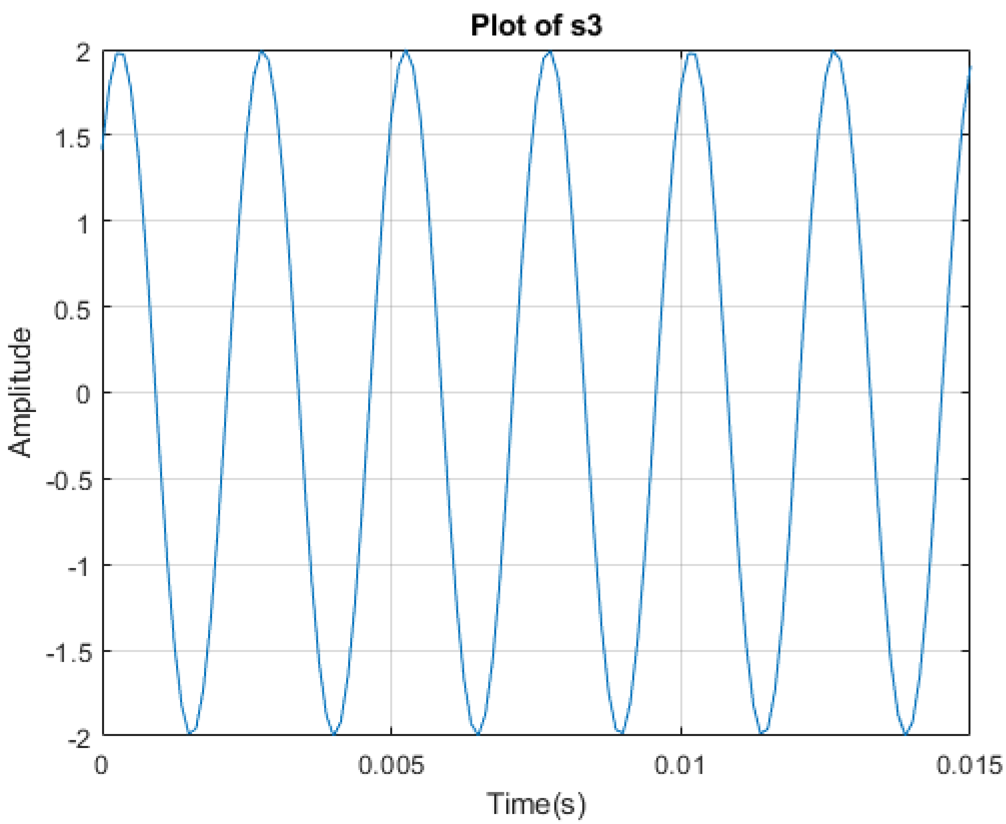
```
%(a)
tt = 0:1/8000:1;

s1 = 3*cos(800*pi*tt-pi/3);
figure;
plot(tt, s1);
xlim([0,0.015]);
xlabel('Time(s)');
ylabel('Amplitude');
title('Plot of s1');
grid on;

s2 = 2*cos(800*pi*tt-pi/4);
figure;
plot(tt, s2);
xlim([0,0.015]);
xlabel('Time(s)');
ylabel('Amplitude');
title('Plot of s2');
grid on;

s3 = 2*cos(810*pi*tt-pi/4);
figure;
plot(tt, s3);
xlim([0,0.015]);
xlabel('Time(s)');
ylabel('Amplitude');
title('Plot of s3');
grid on;

%The frequency of s1 and s2 is 400Hz
%So there are 400 periods per second
%The frequency of s3 is 405Hz
%So there are 405 periods per second


%(b)
s1_scaled = s1/max(abs(s1));
s2_scaled = s2/max(abs(s2));
s3_scaled = s3/max(abs(s3));

audiowrite('s1.wav', s1_scaled, 8000);
audiowrite('s2.wav', s2_scaled, 8000);
audiowrite('s3.wav', s3_scaled, 8000);

%(c)
%soundsc(s1_scaled, 8000);
%soundsc(s2_scaled, 8000);
%soundsc(s3_scaled, 8000);
```

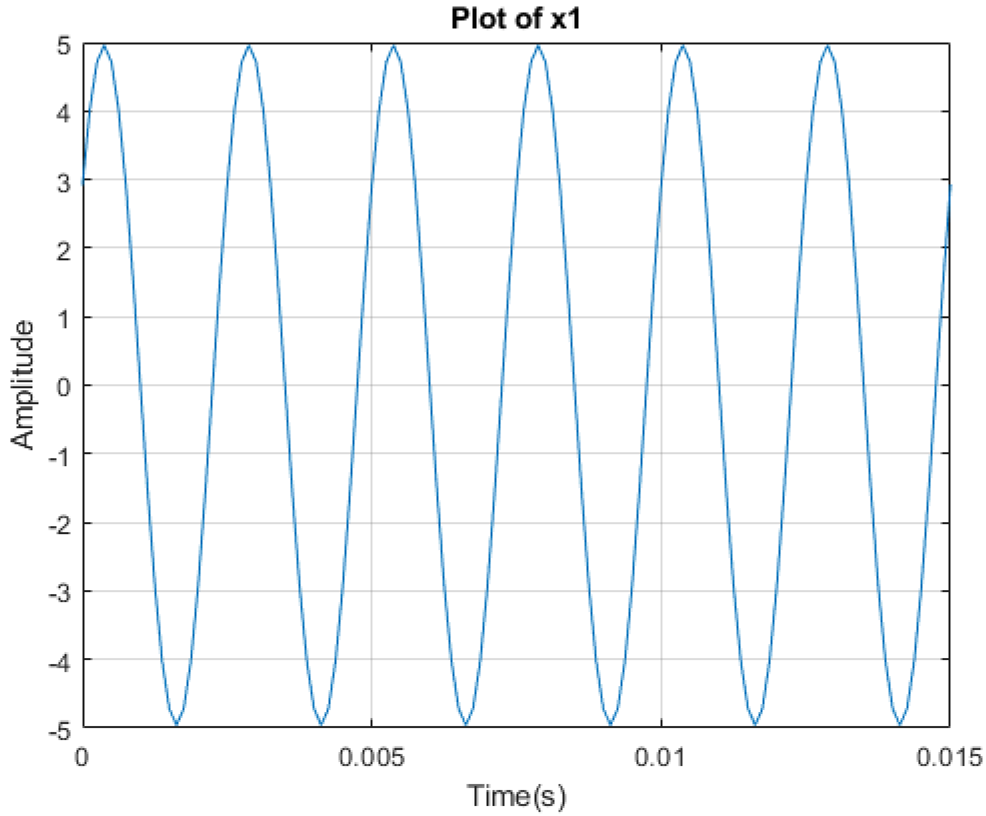
```
%When listening to the generated files,
%they all sound pretty much the same.
%However, s3 sounds a little higher pitch.
%But it is so small it is hard to tell.
%I had to sort of play the files at
%the same time in order to tell.


%(d)
%
% <<Exercise1partd.jpg>>
% I couldn't get this to work. It keeps saying 'Image not found' or
 just
% not being published. I tried different file types, placing the image
 in
% different folders, different synatax. I don't know what else to do.


x1 = s1+s2;
figure;
plot(tt, x1)
xlim([0,0.015]);
xlabel('Time(s)');
ylabel('Amplitude');
title('Plot of x1');
grid on;


%(e)
x2 = s2+s3;
figure;
plot(tt, x2)
xlim([0,0.5]);
xlabel('Time(s)');
ylabel('Amplitude');
title('Plot of x2');
grid on;

%fundamental frequency is the GCD of both frequencies.
%frequency of s2 = 400, s3 = 405
%GCD of 400 and 405 = 5
%with this frequency, the period should be 0.2 seconds. I don't know
 why my
%plot shows a period of 0.4 seconds?

x2_scaled = x2/max(abs(x2));
audiowrite('x2.wav', x2_scaled, 8000);
%soundsc(x2_scaled, 8000);

%x2.wav sounds like it is sort of vibrating. It is not a constant tone
 like
%the other .wav files. This is because the two siunoids that make up
 x2
```

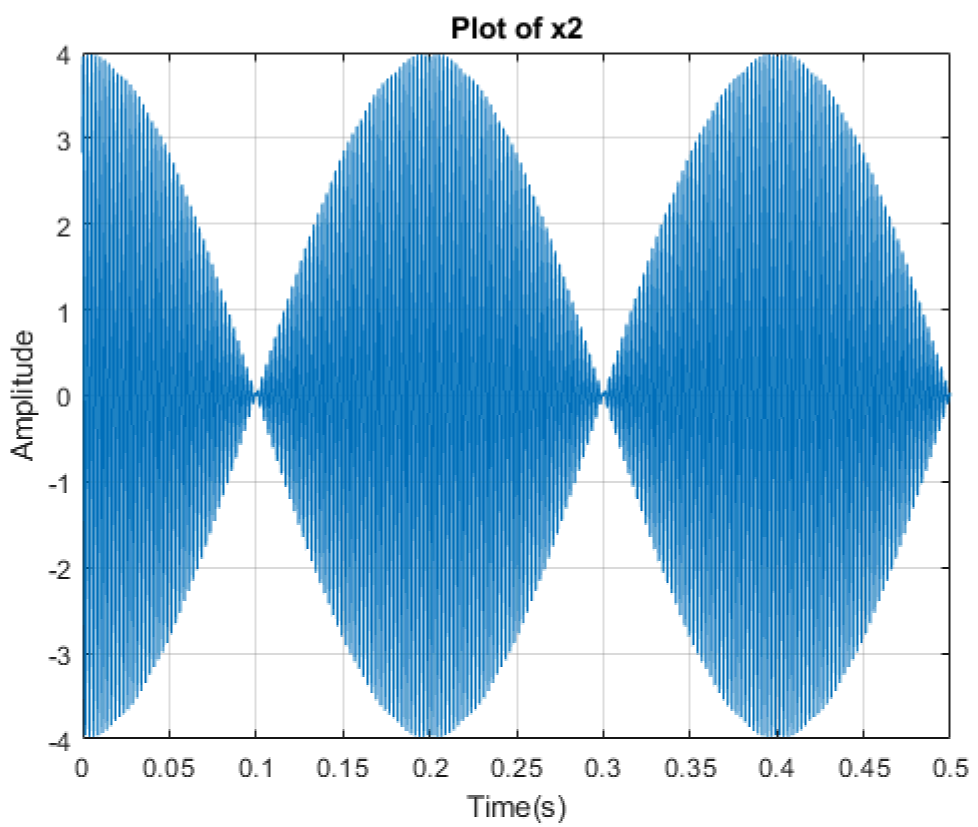
```
%have different frequencies so we are hearing each fequency sort of
 'fight
%each other
```




Plot of s2
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
Amplitude
0
0.005
0.01
0.015
Time(s)


Plot of s3
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
Amplitude
0
0.005
0.01
0.015
Time(s)


Plot of x1
5
4
3
2
1
0
-1
-2
-3
-4
-5
Amplitude
0
0.005
0.01
0.015
Time(s)

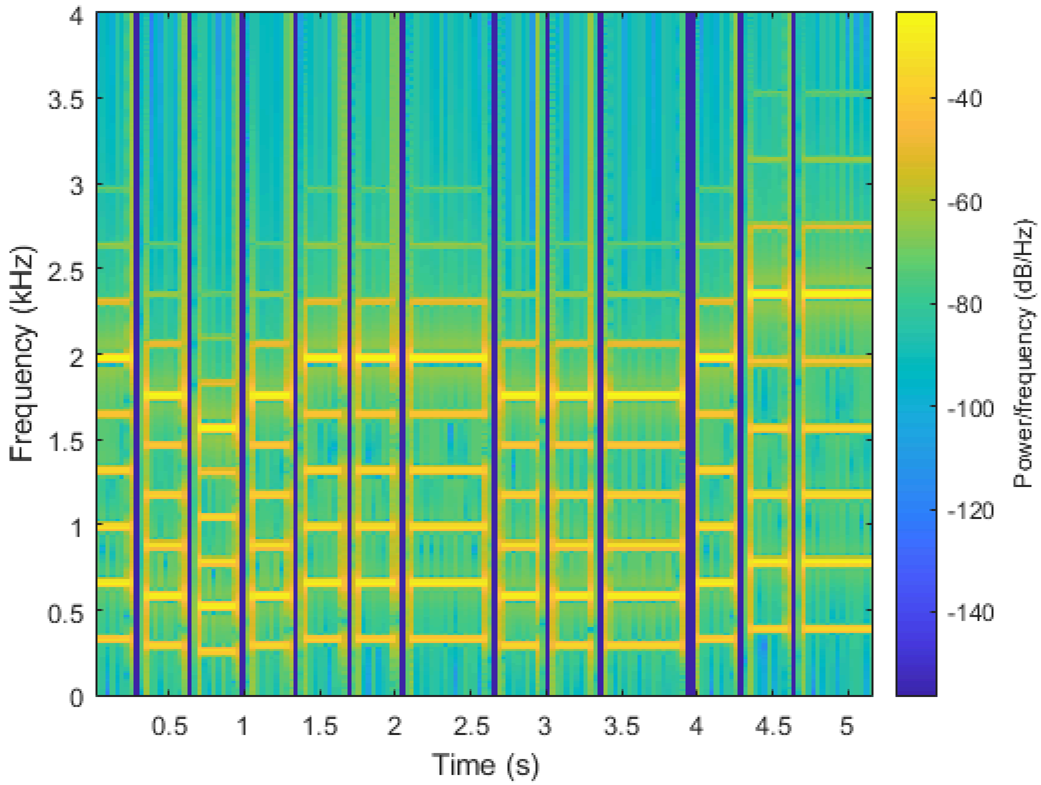

Plot of x2
4
3
2
1
0
-1
-2
-3
-4
Amplitude
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
Time(s)

Exercise 2.2

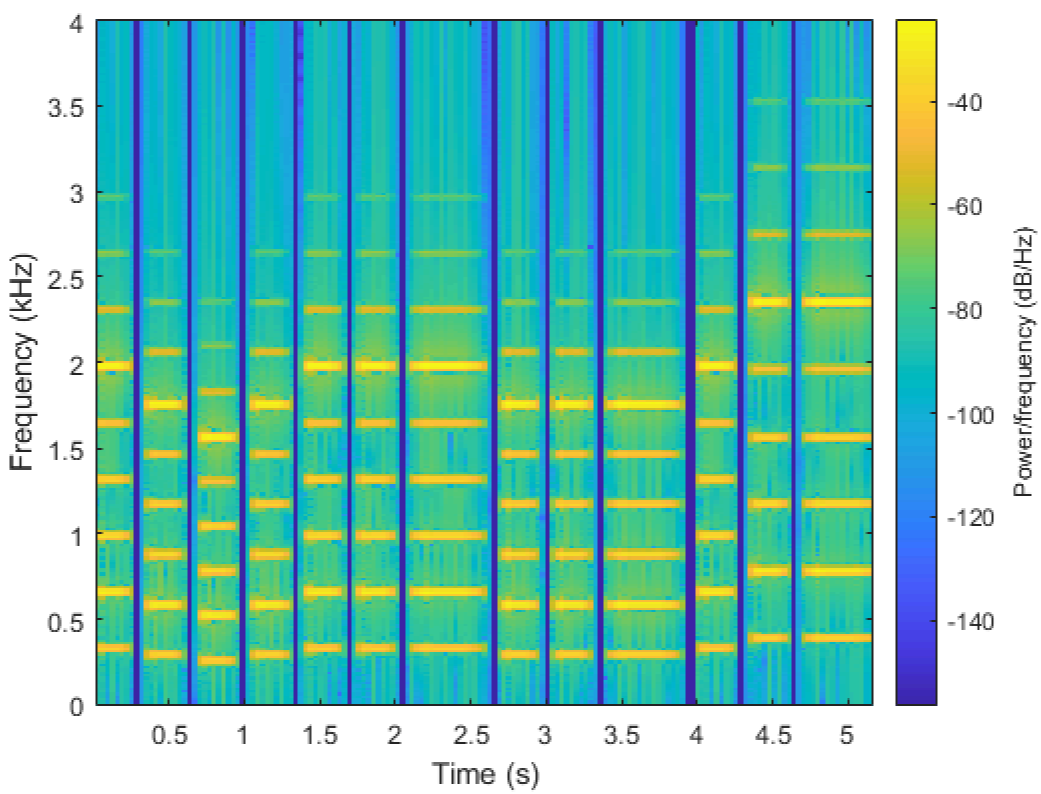
```
%(a)


% <include>key_to_note.m</include>


%(b)
keys = [64 62 60 62 64 64 64 62 62 62 64 67 67];
Xs = [1 1 1 1 1 1 1 1 1 1 1 1 1];
start_times = [0 0.35 0.7 1.05 1.4 1.75 2.1 2.7 3.05 3.4 4 4.35 4.7];
end_times = [0.25 0.6 0.95 1.3 1.65 2 2.6 2.95 3.3 3.9 4.25 4.6 5.2];
fs = 8000;

% <include>build_song.m</include>

x = build_song(keys, Xs, start_times, end_times, 8000);
audiowrite('mary.wav', x, 8000);
%soundsc(x, 8000)
%The notes kind of sound like they are clicking. There is a distinct
 tap at
%the begining and end of each note.


%(c)
spectrogram (x , 512 , 256 , 512 , fs , 'yaxis');
%I honestly don't know what any oddities would be. I do think that the
%purple lines are wierd but that is just the pauses in between each
 note.
%The purple does show to be a very low frequency on the spectrogram so
 the
%immediate switch from that low frequency to our desire frequecy can
 be
%what is makeing the tapping noise.

%(d)

%Here is the code I used to create my phasor vector using the amplitue
 and
%phase
amplitude = [0.1155, 0.3417, 0.1789, 0.1232, 0.0678, 0.473, 0.026,
 0.0045, 0.002];
phase = [-2.1299, 1.6727, -2.5454, 0.6607, -2.039, 2.1579, -1.0467,
 1.8581, -2.3925];

phasor = zeros(1,length(phase));
for count = 1:length(phase)
   phasor(count) = amplitude(count) * cos(phase(count)) +
 j*amplitude(count)*sin(phase(count));
end
```

```
% <include>key_to_musical_note.m</include>


% <include>build_song_2.m</include>

y = build_song_2(keys, phasor, start_times, end_times, 8000);
%soundsc(y, 8000)
audiowrite('mary_trumpet.wav', y, 8000);
spectrogram (y , 512 , 256 , 512 , 8000 , 'yaxis');

%This spectrogram now has 9 sinusoids ploted during each of the notes
 time
%periods. Each sinusoid is also a different shade meaning they are all
%different frequencies. Also each sinusoid seems equidistant from the
 other
%sinudoids above and below it.

%(e)

% <include>adsr.m</include>

figure;
[ note tt ] = key_to_note (69, 1, 0.25, 8000 );
plot(tt, note);
hold on;
plot(tt, adsr(note), 'r -');
hold off;

%The original is just a regular sinusoid. The second plot takes the
 first
%plot but manipulating its amplitude differently over time.


% <include>build_song_3.m</include>


z = build_song_3(keys, phasor, start_times, end_times, 8000);
audiowrite('mary_trumpet_adsr.wav', z, 8000);
spectrogram (z , 512 , 256 , 512 , 8000 , 'yaxis');

%So this spectrogram is very similar to the one obtained in (d) except
 for
%this spectrogram does not have the vertical yellow at the beginning
 and
%end of each note. That would make sense on why the first songs
 sounded
%like they were tapping, but this one sounds smooth. Those drastic
%frequency changes are gone when using the adsr function.
```

Warning: Data clipped when writing file.
Warning: Data clipped when writing file.



Frequency (kHz)
Time (s)
Power/frequency (dB/Hz)



Frequency (kHz)
Time (s)
Power/frequency (dB/Hz)

Published with MATLAB® R2020a