Audio Processing with MatLab
An Introduction

By: Rachel Hager

This lab is an introduction to audio processing with MatLab. This lab will help to familiarize you with some of the main functions to read in and play music files in MatLab.

The following functions that will be used in this lab are:

wavread() sound() soundsc() flipud() wavwrite()

To understand how each of these functions is used in MatLab, type *help* followed by one of the above commands into the command prompt. Note that in order to use the wavread() function you must use a .wav file. It is possible to convert .wma and .mp3 files into .wav files. You can Google search to find programs to do this. If you can't find one that works, send me an email and I can forward you some links.

This lab consists of the following sections:

Section 1: Read and Store an Audio File in MatLab
Section 2: Play the Audio File
Section 3: Audio Scaling
Section 4: Playing a Track Backwards
Section 5: Practice What You've Learned
Section 6: Create Music with MatLab

Let's get started!!!

Section 1: Read and Store an Audio File in MatLab

To read and store an audio file, you can use one of two different command lines. The following stores the file into variable y.

```
y = wavread('filename');
```

Remember to include the entire filename including the directory.
Example: C:\My Documents\EE186Labs\Audio.wav

The command line below stores the audio file into variable y and the sampling frequency in variable Fs.

[y,Fs] = wavread('filename');

Section 2: Play the Audio File

To play an audio file in MatLab you use the sound() function. The following function plays the sound. If the Fs variable is not defined or included in the command, it will assume the default sample rate of 8192 Hz.

sound(y,Fs);

Section 3: Audio Scaling

To scale an audio file the soundsc() command is used. This allows for the modification of an audio signal's amplitude or frequency.

soundsc(y,Fs);

To increase the volume of the audio track you can multiple the variable it is stored in by a scalar. To slow down or speed up the track played you can adjust the sampling rate. Comment on your observations using different values.

Now experiment with different bit values (1,2,..,16) in the following command:

soundsc(y,Fs,bits);

Comment on your observations.

Section 4: Playing a Track Backwards

The command to reverse the order of the samples in a matrix is flipud(). Experiment with this command.

Section 5: Practice What You've Learned

Now for this part of the lab have fun with MatLab. Take your favorite music files and convert them to .wav files if they are not already converted. I'm sure you all have audio files from some source or another!!! If you're not sure how to convert the files, Google search it.

Once you have a couple of your favorite audio files, you may need to crop the file. Most audio files are fairly large which can cause MatLab to lock up.

Read and store the cropped audio file. Now use some of the above commands to modify the audio signal. Play your favorite song backwards or make your favorite singer sound like a chipmunk. Note that the above commands are just a basic look at MatLab's audio processing capabilities.

Comment on your findings!!!

For your help here is a file forward and backward that has been done with the same process

Section 6: Create Music with MatLab

This section of the lab will teach you how to create songs using different tones created in MatLab. First we are going to code a sine wave of amplitude A = 1, with at an audio frequency of 523.25 Hz (corresponds to C).

cnote = sin(2*pi*523.25*(0:0.000125:0.5));

This vector cnote now contains simples of the sine wave from t = 0s to t = 0.5s, in samples that are spaced 0.000125s apart (this is the sampling interval Ts). Note that this sampling interval corresponds to a sampling frequency of 8 kHz (1/Ts = fs). This is standard for voice grade audio channels.

Now to write this sound to a wave file we have the following command.

wavwrite(cnote, 'c.wav');

Now to play the sound, same as before use the sound() function. That is the single note that you just created. Now that you've got one down… only several more to go!!

I've done some of the easy work here for you. The following webpage gives the frequencies of different notes which are also shown on the table below.
http://www.dolmetsch.com/musictheory27.htm

Tuning Pitch: A=440Hz.								
A	27.50Hz.	55.00Hz.	110.00Hz.	220.00Hz.	440.00Hz.	880.00Hz.	1760.00Hz.	3520.00Hz.
A#	29.13Hz.	58.27Hz.	116.54Hz.	233.08Hz.	466.16Hz.	932.32Hz.	1864.65Hz.	3729.31Hz.
B	30.86Hz.	61.73Hz.	123.47Hz.	246.94Hz.	493.88Hz.	987.76Hz.	1975.53Hz.	3951.06Hz.
C	32.70Hz.	65.40Hz.	130.81Hz.	261.62Hz.	523.25Hz.	1046.50Hz.	2093.00Hz.	4186.00Hz.
C#	34.64Hz.	69.29Hz.	138.59Hz.	277.18Hz.	554.36Hz.	1108.73Hz.	2217.46Hz.	
D	36.70Hz.	73.41Hz.	146.83Hz.	293.66Hz.	587.33Hz.	1174.65Hz.	2349.31Hz.	
D#	38.89Hz.	77.78Hz.	155.56Hz.	311.12Hz.	622.25Hz.	1244.50Hz.	2489.01Hz.	
E	41.20Hz.	82.40Hz.	164.81Hz.	329.62Hz.	659.25Hz.	1318.51Hz.	2637.02Hz.	
F	43.65Hz.	87.30Hz.	174.61Hz.	349.22Hz.	698.45Hz.	1396.91Hz.	2793.82Hz.	
F#	46.24Hz.	92.49Hz.	184.99Hz.	369.99Hz.	739.98Hz.	1479.97Hz.	2959.95Hz.	
G	48.99Hz.	97.99Hz.	195.99Hz.	391.99Hz.	783.99Hz.	1567.98Hz.	3135.96Hz.	
G#	51.91Hz.	103.82Hz.	207.65Hz.	415.30Hz.	830.60Hz.	1661.21Hz.	3322.43Hz.	

Using this information create a few different notes in MatLab.

Note that there are different octaves as you go through the different keys on a piano. To get you started here are a few:

```
f = sin(2*pi*174.61*(0:0.000125:0.5));
g = sin(2*pi*195.99*(0:0.000125:0.5));
a = sin(2*pi*220*(0:0.000125:0.5));
b = sin(2*pi*246.94*(0:0.000125:0.5));
```

Now to create a line of music use the following command:

```
line1 = [a,b,c,d,e,f];
line2 = [a,b,c,d,e,f];
```

The letters should represent the notes that you have created in MatLab. Put the notes in the order you want them to play.

To create your song use:

```
song = [line1, line2];
```

Add as many lines as you like. Now write your song, read and store it into a new variable, and listen to it. Comment on your musical abilities. If you're musically challenged, check out Google for some songs that you can create. Feel free to create your own song or attempt to copy some Beethoven. Enjoy!!!

If you need some more help with this lab, check out the following link:

http://users.rowan.edu/~shreek/networks1/music.html

For another example, at the end of this lab I have attached the song that I created which is a few lines of "Heart and Soul". Get creative with this assignment and comment on the problems you run in to.

REFERENCE

http://class.ee.iastate.edu/mmina/ee186/labs/Audio.htm