Laboratory Project M1:
Simple Manipulations of a Music Sample

***Abstract*-You will use Matlab® script files that have been provided to you to manipulate a sample of music that is available in Matlab®. You will also create a simple sound effect.**

I. Preparation

Read *Matlab Primer* pages 1-2 to 1-13. Be sure you **bring your earbuds** to all labs so you can listen to sounds produced by your Matlab® programs.

II. Learning Objectives

1) Learn basics of using script file in the Matlab® workspace.
2) Become familiar with processing of digitized sounds.
3) Hear the effects of distortions of, or noise added to, the music sample.

III. Introduction

Matlab® has the following built-in sounds that may be easily loaded into the Matlab® workspace: chirp, gong, handel, laughter, splat, and train.

We will use the "handel" music sample, a snippet from Handel's *Messiah*. The music is stored as a series of values of the music waveform, which is originally a continuous function of time, taken at regular intervals, namely 8000 times per second. This is a much slower rate than the typical 44,100 samples per second employed by CD players, but it is sufficient for recording intelligible speech and music.

Once you have loaded the "handel" music into Matlab®, you will be able to see what the music waveform looks like, and you will be able to hear what various sound effects sound like. Later on, you will be asked to create your own original sound effect.

IV. Procedure

A. Copy Script Files to Your Account

Open the website for the ECE 1250 course:
http://www.ece.utah.edu/eceCTools/ECE1250/
On the ECE 1250 site, find Lab M1 and "Matlab Scripts for Lab M1." Click on each script file and copy it into a file of the same name in a folder of your choice on your computer. Use a text-only file format, such as that used by Notepad, and make sure each file has a ".m" extension so Matlab will recognize it as a Matlab® file. These script files contain simple Matlab® commands to manipulate the "handel" sound snippet. (You can use these files with other sounds available in Matlab®, too.)

When you are finished, you will have the following script files: ah.m, ay.m, chop.m, commands.m, echoes.m, highpass.m, listen.m, loadhan.m, louder.m, lowpass.m, noise.m, oo.m, softer.m, spectrum.m, speedup.m, warm.m, and waveform.m.

B. Start Matlab®

Login to your computer and double click on the Matlab® icon on the desktop to launch the Matlab® application.

Under the *File* menu, click on *Set Path* and use the *Add* button to add the folder into which you put the Matlab® script files. This will allow Matlab® to locate your script files. Otherwise, Matlab® will output a cryptic error message to the effect that you have entered an undefined variable or function. Once you have added your folder to the list of folders through which Matlab® will search, be sure you Save the new configuration. Then click on Close.

If *Set Path* is unavailable because of write permission problems, use the current folder window to navigate to the folder where you have stored the Matlab® script files.

Now you are ready to listen to sounds.

C. Load "handel" Sound and Listen to it.

To see a list of commands available to you and to verify that your script files are working, enter the following command at the >> Matlab® prompt. After typing the command, hit the *Enter* key:

```
>> commands
```

You should see a list of commands, each of which consists of single word. (The // and what follows is a comment and is not part of the command.) Use this command whenever you want to see a list of available commands.

Now you are ready to listen to sounds. Type the following command (followed by the *Enter* key, which completes any command typed at the Matlab® prompt):

```
>> loadhan
```

You have now loaded a clip from Handel's *Messiah*. To listen to it, plug your earbuds into the green-ringed output on the front of the compute (fold down the small panel at the bottom front of the computer, if necessary), and use the waveform command:

```
>> listen
```

Any time you want to listen to a sound, use the listen command.

To see the waveform itself, use the "waveform" command. It will be easier to see what the actual waveform looks like if you are working with a shorter snippet, such as "ah", "ay", and "oo".

D. Experiment with the Sound Commands

Use the >> diary on command to record your commands for this section.

Use the various commands that load, plot, or alter a sound snippet to become familiar with how a sound may be modified by digital signal processing. Note that commands are cumulative. Two "softer" commands in a row will lower the sound volume twice, for example. When you wish to start afresh, use the "loadhan" or "ah", "ay", or "oo" commands.

E. Create a Sound Effect Script File

Experiment with the sound processing commands until you find a combination of commands that produces an interesting sound. In the folder containing the Matlab® command files you created at the outset of this lab, use Notepad or a similar plaintext editor to create a file called *name*`_sound.m` (where *name* is replaced with your first initial and last name) and list the commands that produce your interesting sound. This file will have one command per line with no blank lines and no spaces at the beginning of each line. For example, the following file called `NCotter_sound.m` would play a low-passed version of the "handel" snippet:

```
loadhan
lowpass
listen
% Neil Cotter's sound effect
```

The last line of your command file must be a comment (which starts with a % symbol) that includes your name, as in the above example.

F. Create a New Command

Look at what is in some of the command files you created at the outset of this lab and mimic them to create your own sound-altering command. That is, create your own command file. Name your command file *name*_command.m (where *name* is replaced with your first initial and last name). **When you first try your new command, set the computer's output volume at a low level and hold your earbuds away from your ear to avoid a loud sound.** If the sound is too loud, multiply the sound variable, *y*, by a constant less than 1.0, as in the "loadhan" command. Note that the sound played by "listen" must be in a variable called *y*.

For a list of elementary functions in Matlab® that you might want to use for this last part of the lab, type

```
>> help elfun
```

for a list of elementary functions.

Another idea would be to modify the spectrum of the sound in some way. The spectrum is roughly equivalent to an array of phasors that describe the sinusoids that, when added together, create the sound waveform. The frequencies of the phasors in the spectrum start at 0 Hz (DC) and and are equally spaced, ending at 8000 Hz. (Actually, the last 8000 Hz entry in the spectrum array is left out.)

The fft() command computes the spectrum, and the ifft() command converts the spectrum back into a sound. Here is some code with comments that shows how one might modify the spectrum of the sound by simply zeroing out certain frequencies:

```
s = fft(y);     % Computes the spectrum of y.  Result is complex.  Spectrum command
                % shows the frequency content from 0 Hz to 8000 Hz in the signal.
                % x-axis of plot has same number of pts as y.  No matter how many
                % samples there are on the x-axis, they correspond to 0 Hz to
                % 8000 Hz.
s(2000:4000) = 0;    % zeros out part of the spectrum
y = real(ifft(s))    % Converts spectrum back into sound.
                % ifft = inverse fast Fourier transform
                % The sound must be real-valued, so we take real part of sound.
```

You may also wish to try applying some elementary functions to `s`. You might stumble onto a very interesting sound. When you have completed your command file, play your sound for the TA.

G. What to Turn In

Your TA will instruct you on what format to use when turning in your work. Typically, you will email him or her your files. For this lab, turn in the following:

1) Your diary file for part *D*
2) Your *name*_sound.m file for part *E*
3) Your *name*_command.m file for part *F*.