Laboratory Assignment 2: Signal Sampling, Manipulation and Playback

**Purpose**

In this Lab, you will learn how to record sounds and save them in MATLAB. You will learn how to perform mathematical modifications to digital audio signals in order to produce some interesting audio effects.

**Objectives**

By the end of this laboratory assignment, you should be able to:

* Record sound in MATLAB
* Use some simple mathematical manipulations to alter audio properties

**Reference**

Review Topics:

* Mathematical representations of continuous- and discrete-time signals
* Mathematical operations on signals
* Representation of sinusoids at specified frequencies

1. **Record and save a speech signal**

* Record yourself saying “hello world.”
  + Make sure the microphone is on.
  + In the Matlab workspace window, type
    - fs = 10000; %this is the sampling frequency in Hz
    - s= audiorecorder (fs, 16, 1); %16 bits, one channel
    - recordblocking(s, 4); % start Recording for 4 seconds
    - speech = getaudiodata(s, 'int16')
* Save the speech you just recorded
  + audiowrite('myspeech.wav',speech,fs); % creates a .wav file of the signal speech

1. **Read and play the speech you just recorded**
   * [sp,fs] = audioread('myspeech.wav'); % reads in a signal and the recording frequency, fs,
   * soundsc(sp, fs) %plays the read signal at frequency, fs
   * Try playing the signal at different frequencies such as fs/2 and 2fs (note what happens and explain why)
2. **Plot and observe your recorded voices**
   * Plot the Waveform
     + plot(speech);
   * On your own add a title, axes labels and grid to the plot
3. **Calculate the pitch of each member of your group. Determine if this seems appropriate**
4. **Some Audio Effects**

Apply each of the following MATLAB functions to your speech file, ‘my speech’. Save the new signal to a .wav file using 'wavwrite' and then play back the signal. Observe the effect of these digital manipulations on the audio properties of the signal.

1. The MATLAB function **half** created by you in the first laboratory.
2. The MATLAB function **double** created by you in the first laboratory,
3. The MATLAB built-in functions **fliplr** and **flipud.**

Document how the MATLAB functions you use change the properties of the original tone and speech signals. In each case, does the speech segment sound like it was produced by the same speaker as the original? Why and/or Why not.

1. **Echo Effects**
2. Develop a MATLAB function, ‘myEcho’ to generate an echo. The output of this function should be the echo signal and the signal that you hear. A simple analytical model of an echo se(t) of a signal s(t) is se(t) = α(t-T)s(t-T), where the α (t) function attenuates the volume over time and T is the time delay introduced by reflection. In this treatment, we will assume that 0 ≤ α(t) < 1 and T> 0. This implies that the echo's strength is weaker than the true signal and that the echo always arrives at a time later than the true signal (post-echo). The signal r(t) that you hear is formed by a simple addition of the original signal and its echo.
3. Use the MATLAB function file you created to create an echo signal using your original speech signal and a constant attenuation. You will need to create a signal that is a scaled version of the original and add it to the original with the desired offset in time. Experiment with different values for the time delay and attenuation. Listen to the resulting signals and note your observations.
   1. Generate a 0.25-second echo effect. Let α=0.65.
   2. Use a non-constant attenuation function, such as an exponential: α(t)= Ae-t/τ . Determine values of A and τ that make an interesting effect.
   3. Use several different delays and amplitudes to get a variety of echoes.
   4. Use an oscillatory attenuation function, such as α (t)=Acos(ωt). Try different values for ω.

*Describe the impact that your choices of Time delay and attenuation α have on the resulting synthesized echo. Be specific out changes in parameters and relative impact on sound. What happens if T changes during the echo-generation process?*