Fall 2017 EE 313 Linear Systems and Signals Prof. Evans

Mini-Project #2: Digital Audio Processing

Assigned on Friday, October 20, 2017
Due on Friday, October 27, 2017, by 12:30 pm via Canvas submission

Late mini-projects will not be accepted.

Reading: McClellan, Schafer and Yoder, *Signal Processing First*, 2003, Chapters 1-6.
Companion Web site with demos and other supplemental information: http://dspfirst.gatech.edu/

For the contact information for the TA, please refer to

https://utexas.instructure.com/files/43585083/download?download_frd=1

Office hours for Ms. Ghosh and Prof. Evans follow, as well as Prof. Evans' coffee hours on Friday.

Time Slot	***Monday***	***Tuesday***	***Wednesday***	***Thursday***	***Friday***
9:00 am			**Ghosh**		
9:30 am			**Ghosh**		
10:00 am			**Ghosh**		
10:30 am					
11:00 am		**Ghosh**		**Ghosh**	
11:30 am		**Ghosh**		**Ghosh**	
12:00 pm		**Ghosh**		**Ghosh**	Evans (EER cafe)
12:30 pm		Evans (EER 1.516)		Evans (EER 1.516)	Evans (EER cafe)
1:00 pm		Evans (EER 1.516)	Evans (EER 6.882)	Evans (EER 1.516)	Evans (EER cafe)
1:30 pm		Evans (EER 1.516)	Evans (EER 6.882)	Evans (EER 1.516)	Evans (EER cafe)
2:00 pm			Evans (EER 6.882)	Evans (EER 6.882)	
2:30 pm				Evans (EER 6.882)	
3:00 pm				Evans (EER 6.882)	

You may work individually or with one partner. If you work with a partner, then create one report together. Each of you would submit the same report on Canvas. Be sure that the mini-project report represents the independent work of the author(s) on the report.

Please use headphones for this miniproject for the sanity of those around you.

1.0 Digital Audio Input and Output

There are many ways to create a digital audio signal, including recording from a microphone, reading from a file and generating sound from a formula. Each of these ways is explored in a different subsection below. To check to make sure that the digital audio signal has been correctly recorded, read or generate, the subsections describe how to play back and plot audio signals.

1.1 Recording from a Microphone

Use the following MATLAB code to record sound from your laptop’s microphone:

```
% Record from the microphone
fs = 44100;
numBits = 16;
numChannels = 1;
recordingTime = 5;
recObj = audiorecorder(fs, numBits, numChannels);
disp('Start recording...');
recordblocking(recObj, recordingTime);
disp('End recording.');

% Store data in double-precision floating-point array
myRecording = getaudiodata(recObj);

% Play back the recording with automatic scaling
soundsc(myRecording, fs);

% Plot the waveform in the time domain
figure;
plot(myRecording);

% Plot the spectrogram
figure;
spectrogram(myRecording, hamming(16384), 8192, 16384, fs, 'yaxis');
```

Submit the time-domain plot and spectrogram of a recording from your microphone. For the time-domain plot, what is the average value of the signal? You can use the MATLAB command `mean` to validate your observation. Are there any quite portions in the recording? Please circle the quiet portions in the time-domain plot. From your observation of the spectrogram, please describe any principal frequencies or other patterns.

1.2 Reading from a File

Use the following MATLAB code to read and playback an audio file. The `audioread` command in MATLAB supports several audio formats such as MP3 and WAV:

```
waveFilename = ‘file.ext’;             % Pick an audio filename
[y,Fs] = audioread(waveFilename);
soundsc(y, fs);
spectrogram(y, hamming(16384), 8192, 16384, fs, 'yaxis');
```

One can specify the number of samples to read from a file in the second argument to `audioread`. Without a second argument specified, the entire file is read.

Submit the spectrogram of the audio file that you read in. From your observation of the spectrogram, please describe any principal frequencies or other patterns.

1.3 Generating Sound from a Formula

We can generate sinusoidal tones, linear sweeps (chirps), and other signals through formulas.

```
% Sampling rate and time
fs = 44100;
Ts = 1/fs;

% Generate a chirp signal
tmax = 5;
t = 0 : Ts : tmax;
fstart = 20;
fend = 4000;
y = chirp(t, fstart, tmax, fend);

% Play back the chirp signal
soundsc(y, fs);

% Spectrogram
spectrogram(y, hamming(16384), 8192, 16384, fs, 'yaxis');
```

Please submit the spectrogram, and describe the pattern you see.

2.0 **Digital Audio Processing**

We can apply a variety of audio effects using finite impulse response (FIR) filters.

Please complete *Section 3.3 Filtering a Music Waveform* in the following laboratory assignment from *DSP First* at

http://dspfirst.gatech.edu/chapters/DSP1st2eLabs/SamplingConvFiltering.pdf

For the music signal to process, please record a signal using the microphone as explained in Section 1.1 in this document.

In addition, please submit plots of the impulse response (using `stem`) and frequency response (using `freqz`) of the FIR filter you designed in part (a) of *Section 3.3 Filtering a Music Waveform*. Please describe the impulse response as well as magnitude and phase responses of the FIR filter.

3.0 **Saving an Audio Signal to a File**

An audio signal can be saved to a file as follows. The `audiowrite` command in MATLAB supports several audio formats such as MP3 and WAV:

```
% Sampling rate and time
fs = 44100;
Ts = 1/fs;

% Generate a chirp signal
tmax = 5;
t = 0 : Ts : tmax;
fstart = 20;
fend = 4000;
y = chirp(t, fstart, tmax, fend);

% Save the audio signal to a file
waveFilename = ‘chirp.mp3’;
audiowrite(waveFilename, y / max(abs(y)), fs);
```