Computer Assignment 1

For this assignment, you will be uncompressing audio clips that I have compressed using compression algorithms of my own invention. The compression algorithms works by breaking the audio clip into *windows*, and then representing each window as a sum of cosines. You will be given data that correspond to a list of frequencies and their associated complex A values. Using these data, you will uncompress the audio by rebuilding the signal out of cosines.

I used two different methods to try compressing the audio. The steps I did are as follows. Note that in all cases, the sampling frequency is $F_s = 44.1$kHz.

Method 1

- First I divided the signal into 50ms windows (which corresponds to 0.050 seconds/window $\times$ 44100 samples/second = 2205 samples/window)
- Then for each window, I determined the top 30 energy-containing frequencies. For each window, I stored those 30 frequencies (in Hertz) along with their corresponding 30 complex `A` values.

Method 2

- First I divided the signal into 50ms windows (which corresponds to 0.050 seconds/window $\times$ 44100 samples/second = 2205 samples/window)
- Then for each window, I found the energy at 30 different frequencies, evenly spaced between 100 and 2000Hz. The same 30 frequencies were evaluated in each window. For each window, I stored the corresponding 30 complex `A` values. I also stored the 30 frequency values.

I have compressed two different audio files using each of the two Methods. The resulting Matlab data files are:

- `audio_1_method_1_no_overlap.mat`
- `audio_1_method_2_no_overlap.mat`
- `audio_2_method_1_no_overlap.mat`
- `audio_2_method_2_no_overlap.mat`

Your goal will be to reconstruct audio from all four files.

To get started, you can go to Matlab and type `load('audio_1_method_1_no_overlap.mat')` and you should see the following variables in memory:

`fs`: the sampling rate, in Hertz

`freqs`: a matrix of dimension `[200,30]` that stores the frequencies (in Hz) of the strongest 30 harmonics in each of 200 time windows

`A`: a matrix of dimension `[200,30]` that stores the complex coefficient of the various cosines indicated in `freqs`.

Remember that because each window corresponds to 50ms, and because in this case there are 200 windows, your reconstructed audio clip should be $0.050 \times 200 = 10$ seconds long.

The data files for the Method 2 signals contain the same variables except that `freqs` is just a simple 30-element vector, not a fancy 2-dimensional array like in Method 1. That's because in Method 2, its the same frequencies being sampled in every time window - we don't need to specify frequencies for each window because they're always the same.

To rebuild the audio, you will have to create a signal that is 2205 × `nWindows` samples long. For each window, you will need to create and sum the cosine waves that correspond to the frequencies, amplitudes, and phases stored in `freqs` and `A`. Each window should produce a set of 2205 samples; put all those sets end-to-end and you'll have the reconstructed audio signal. Your reconstructed audio clip won't sound great, but you should be able to understand or even recognize what you're hearing. You can play back your audio using `soundsc(x,fs)`. You should hear about 10 seconds of audio in each case.

Honors Students In addition to the above work, students in the Honors section should also reconstruct the audio in the following four files:

- `audio_1_method_1_w_overlap.mat`
- `audio_1_method_2_w_overlap.mat`
- `audio_2_method_1_w_overlap.mat`
- `audio_2_method_2_w_overlap.mat`

For these signals, the compression methods work the same except that each 50ms window overlaps the previous one by 45ms. So if window #1 covers $0 < t < 50$ms, then window #2 covers $5 < t < 55$ms and so on.

Questions - All of the following should be addressed in your discussion section: Which compression method sounds better? Why? Which audio signal sounds better when reconstructed? Why? (think about how the qualties of the audio signals fit with the way the compression algorithms work.) What is the compression ratio for each Method?

What to hand in - Using the provided template, you should hand in a single page report written in MS Word with the following sections: Intro, Methods, Results, Discussion. Your report must answer all of the above questions as well as any other interesting observations you make along the way. You should decide for yourself what information is important to present in your report. The purpose of the report is to convince me that you understand the assignment, so think about the most efficient way of getting your point across. This is a technical report, so avoid humor and colloquial jargon!

Teams - You may work by yourself or in groups of two. If you work in groups, put both your names on the paper, but only one of you needs to do the submission in Canvas. In all cases, you and/or your group may work with other individuals or groups, but you/your group's code and paper must be uniquely yours.

You should also turn in your code in an `.m` file that I can run directly in Matlab. The code should use meaningful variable names and be well commented and otherwise easy for me to read. No need to include the `.mat` files - I already have those.

Zip together the MS Word report (not a PDF) and the Matlab code and submit *a single zip file* through Canvas. You may work in teams of two or by yourself. Submissions should arrive by Friday 1/31/2020 at 11pm.

Useful Matlab Commands

- `size`
- `subplot`
- `xlabel`
- `ylabel`
- `title`
- `xlim`
- `ylim`
- `legend`
- `figure`
- `clf`
- `close`

You can read about all of these by typing `help` and then the command name at the Matlab command prompt.