ENEL420-S2 **Assignment #1** Richard Clare
Version 1.0 1.7.2021

Assignment due: Report and code are due **5 p.m. Friday 20th August.**

Assignment worth: **15%** towards the final grade.

Groups: The assignment is to be completed by pairs of students, and you can choose your partner. If you have found a partner, both of you should use the ***Group self-selection for Assignment 1*** facility on Learn. If you cannot find a partner, email Richard richard.clare@canterbury.ac.nz and he will help you find one.

Description:

1. The assignment is about the detection of additive interference (noise) on an audio signal through spectral analysis, and the design and implementation of ***notch*** filters to remove this interference, and consequently decode the audio signal.
2. Each group will be given their own data file, which comprises an audio signal corrupted by an interference signal. The audio signals and noise are different for each group.
3. A set of tasks are set below. Use MATLAB or Python/SciPy to complete these tasks, except for Task 4 which should be done analytically. Record all the code you create to get your results as script and/or function files. If you use toolbox or scientific library functions, make sure that you understand what those functions are doing.
4. Complete your report with an abstract, introduction, discussion of the tasks, conclusion and references.

Tasks:

1. Plot the sampled time domain audio signal with the time axis in seconds (the signal is unitless).
2. Calculate the spectrum of the signal, and plot the magnitude of the spectrum versus frequency in Hz. Identify the interference signal.
3. Analytically design (i.e. derive using equations) a biquad IIR 2-pole notch filter to remove the interference. A single pair of conjugate zeros and a corresponding pair of conjugate poles is sufficient for the notch [1]. Draw the realization of your IIR filter to remove the interference.
4. Plot the magnitude and phase of the frequency response of your IIR filter, the time domain signal after filtering, and the spectrum of the audio signal after filtering.
5. Design three FIR filters (window, optimal, and frequency sampling) in order to remove the interference, limiting the number of coefficients to 1000 in each case. Document any trade-offs in the design of each filter. Plot the magnitude and phase of the frequency response of your FIR filters, the time domain signals and spectra of the audio signal after filtering.
6. Estimate the SNR (in dB) of the data in the audio file.
7. What quote is in the audio file? What actor(s) are talking and in what movie? Compare the results of your four filters from tasks 4 & 5. Choose which filter is *best* and explain your decision.

Data files:

The signals are saved as a .wav file. Each can be loaded with MATLAB using `audioread` which also returns the sampling frequency:

[Y, FS]= audioread('groupN.wav')

In Python, using the SCIPY toolbox, the signal and sampling frequency can be read by:

```
from scipy.io import wavfile
samplerate, data = wavfile.read('groupN.wav')
```

Code:

Submit to Learn a single zipped file (one for each group) containing all the code files (script and function files) that your group has used to generate your results. You need to explain in the report why specific functions have been used and show how specific argument values have been arrived at.

Please note that it is not necessary to use highly sophisticated functions for this assignment. It is possible to complete the assignment using only a handful of basic MATLAB (Python) functions, e.g. `audioread`, `filter`, `fir1`, `firpm`, `conv`, `freqz`, `fft`, etc., to complete the tasks.

Report:

Your report should be no more than 12 pages total in length (excluding contents and title pages), using a font no smaller than 11-point. Present a description of the process illustrated only by key lines of code and abstracts of functions. Show in the report only the key figures illustrating your filter designs and results for the tasks.

Marking:

Task 1	5%
Task 2	5%
IIR design (Task 3 & 4)	20%
FIR design (Task 5)	20%
Task 6	10%
Task 7	10%
Overall report quality (introduction, conclusion, references, layout, grammar etc.)	15%
Code quality (organization, proper use of functions, commenting)	15%
Total	100%

Refs:

[1] Ifeachor, E.C. and Jervis, B.W., *Digital Signal Processing: A Practical Approach*, Prentice Hall, 2nd Edition, pp 459-62.